SCIENCE NEEDS MAKERS

HOW CITIZEN SCIENTISTS ARE PUSHING THE BOUNDARIES OF HUMAN KNOWLEDGE

NAILS

Repurpose spiky steel for your makes

CRYSTAL RADIO

Build a retro audio system

CLOCKS

Uncover the mysteries of timepiece conservation

FREECAD PICO **T-SHIRTS** DJ-ING

Welcome to HackSpace magazine

Science is the pursuit of knowledge, and there are lots of different bits of knowledge that a scientist can pursue. Some scientists sit in laboratories and run experiments. In these lab-based experiments, they can control the variables and really dive into the mechanisms of how different things work. While this is valuable knowledge, it's also missing a bit of the messiness of the real world. Lab-based experiments don't always sync with what happens in the wider environment.

The problem with running experiments in the real world is that it's big and diverse. Getting meaningful results means getting lots of data. That's where we, as makers, can come in. We can take measurements, set up apparatus, and generally check on the little bit of the world that we're in. Fortunately, there are lots of us. If every maker does just a little bit of science, then the total amount of science done is a lot. Between all of us, we can help understand our environment, the cosmos, and even what's happening at the centre of the Earth. Individually, we can't record or analyse much, but together we can do a lot.

BEN EVERARD

Editor ben.everard@raspberrypi.com

Got a comment, question, or thought about HackSpace magazine?

get in touch at
hsmag.cc/hello

GET IN TOUCH

 hackspace@
raspberrypi.com

 hackspacemag

 hackspacemag

ONLINE

 hsmag.cc

Available on the App Store

GET IT ON Google Play

PAGE **44**
FREE PICO
WHEN YOU
SUBSCRIBE

Contents

108

Cover Feature

CITIZEN SCIENCE

Makers, the world needs you: there's a load of science out there and only you can help

34

Tutorial

Laser cooling

90 Reduce the number of fires in your workshop with frozen drinks cans

112

110

102

82

06

In the workshop
Making T-shirts

66 Add colour and shapes to otherwise dull garments

Interview
Anna Ploszajski

52 You don't really know materials until you know making

20

Concrete Macro Keyboard

By **TheSuperSewcio.**   **hsmag.cc/ConcreteKB**

Twisting our wrists into awkward shapes to access key combinations is torture on the carpal bones. So why not use a macro keyboard for your most commonly used commands? We've looked at a few in these pages, but this is the first one we've seen to feature concrete, which we can only imagine gives the build a weighty, solid feel. Couple that with clicky mechanical key switches and you've got a device that should last forever. ▫

Right
The Romans used concrete. If they'd had keyboards, they would have liked this

Dreamer PCB

By **Boldport**   hsmag.cc/Dreamer
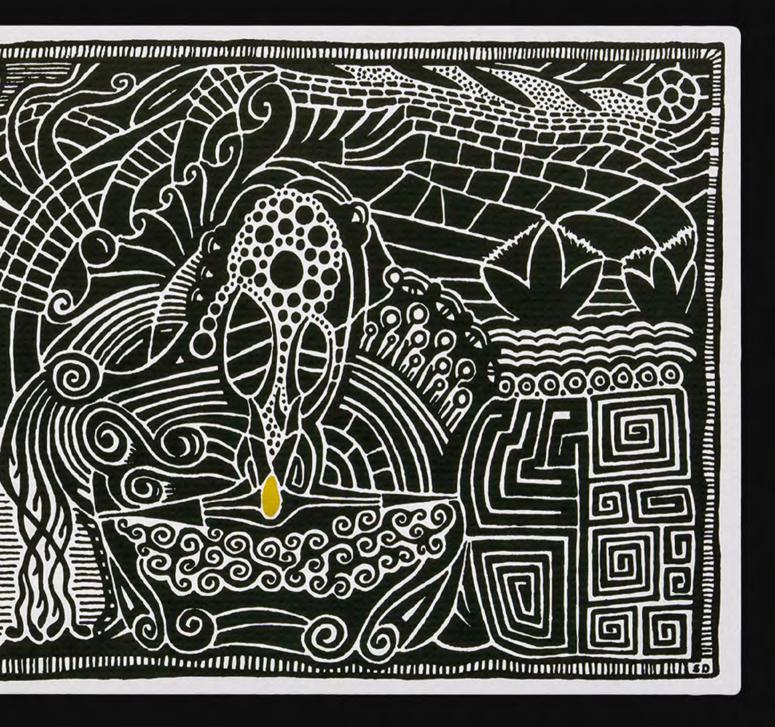
T his is a doodle on a PCB, by serial creator of
beautiful things, Saar Drimer. Unlike most of his
other creations, this doesn't do anything – it just
looks nice. Saar's process begins with pen and ink
doodles, which he scans, modifies, and imports into
PCB software. ◻

Right ↗
Our inner art
historian tells us
that Saar is playing
with the innate
utilitarianism of
the printed circuit
board, and playfully
inverting that by
having the design not
do anything

Pomodoro cube

By **alch_emist**   ◈ **hsmag.cc/PomTimer**

Millions of time-poor, attention-poor individuals can attest to the Pomodoro time management technique. You set a timer, say for 25 minutes, and work your socks off on one task at a time. Then for five minutes, you do something else – put the kettle on, play with the dog, anything that isn't work. When that time's up, it's time to get down to another 25 minutes of uninterrupted concentration.

It's a technique that works, especially for home workers who have difficulty setting boundaries, or who are assailed by Skype, email, Slack, IRC, Twitter, Facebook, and all the other annoying means of communication that flesh is heir to. This build is a fresh implementation of a Pomodoro timer, using an Arduino Nano 33 IoT, a dual seven-segment display, and two strips of NeoPixels that flash when it's time to change work state. Finally, translucent acrylic sheets form the body of the build, and also diffuse the light from the LEDs. ◻

Right ↗
The maker wanted to build something to make study more productive

RC boat

By **Electrosync**   ⌁ **electrosync.com.au**
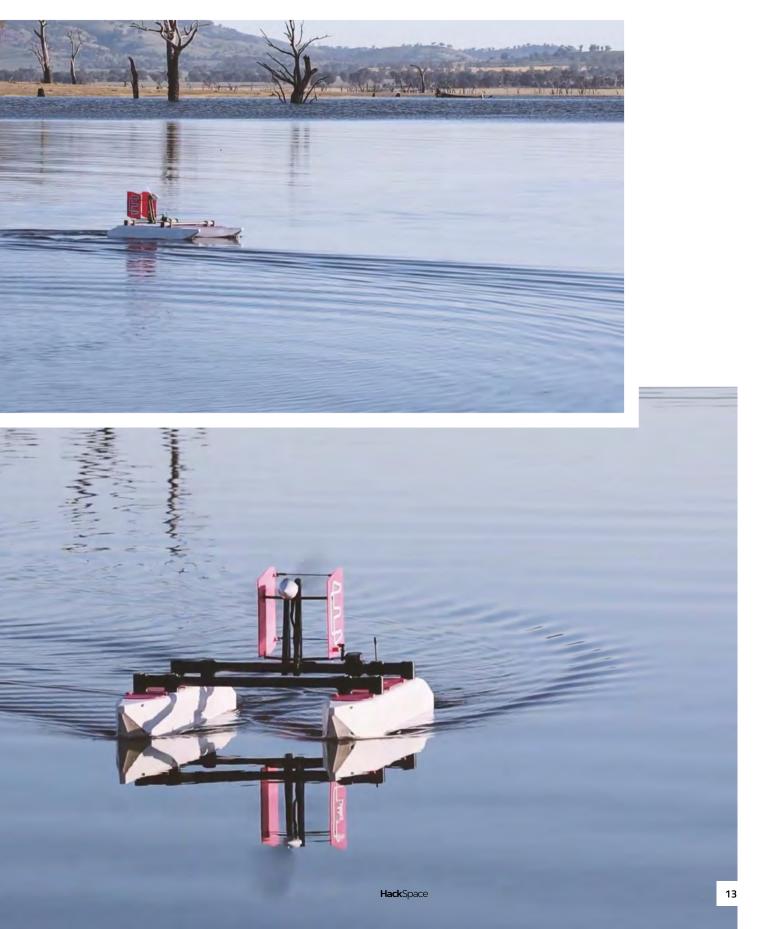
In the UK, we can take water samples in nature pretty **safely.** But in Australia, where everything wants to kill you, there could be a crocodile hiding in every puddle. That's why, if you want to monitor water quality down under, you need a remote-control boat. This one, built by Electrosync, features built-in water sampling and can carry up to 5 kg of water or other sensing equipment. ▢

Right ↗
The boat's construction is a mixture of 3D printing and carbon fibre

Wooden spoons

By **Anna Ploszajski**   ✈ annaploszajski.com

These spoons, carved by materials scientist **Anna Ploszajski, are composed of a polymer of lignin, cellulose, and hemicellulose, a substance with flexibility and high tensile strength.** It's a good thermal insulator, carbon-neutral, and infinitely renewable.

We jest, of course, but tech doesn't always have to be high-tech. There's a load of good science involved in why we choose to make things the way we do. Take a look at page 52 for a deeper dive into this. Or, for a more practical investigation, why not carve your own spoon? ▫

Right ↗
At a push, you can even use this for fuel

Eli's WeatherClock

By **The Digital Vagrant**   ✈ **hsmag.cc/WeatherClock**

W e had a great time this issue speaking with Matthew Read who, among other things, conserves old clocks. One of the things we took away from that Meet The Maker (turn to page 20 to read the whole thing) is that, if you love old objects and want to get an understanding of how they're made, the best thing you can do is build a new thing. Take a Raspberry Pi, take some gears, and make something no one's seen before.

This clock, by The Digital Vagrant, is a perfect example of this philosophy: it's a new version of a weather clock, something that we've seen loads of recently. Rather than the twelve numerals around the outside of the clock, the maker has used weather icons corresponding to that time's conditions. Touch the weather icon for a detailed breakdown for that time, then touch the screen again to return to the clock display. ◻
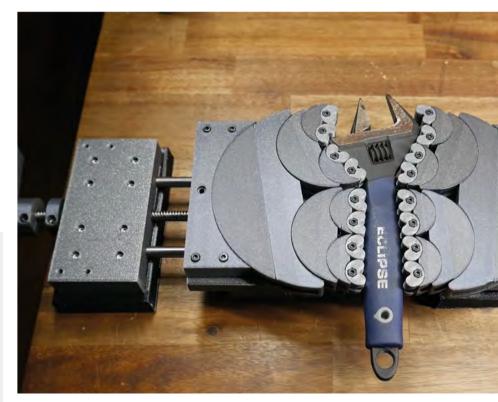
Day	Sunday
hour	9 PM
temp	8.3° C
Feels like	8.3° C
POP	8 %
Rain	--
Wind	0.99 km/h

Objet 3d'art

3D-printed artwork to bring more beauty into your life
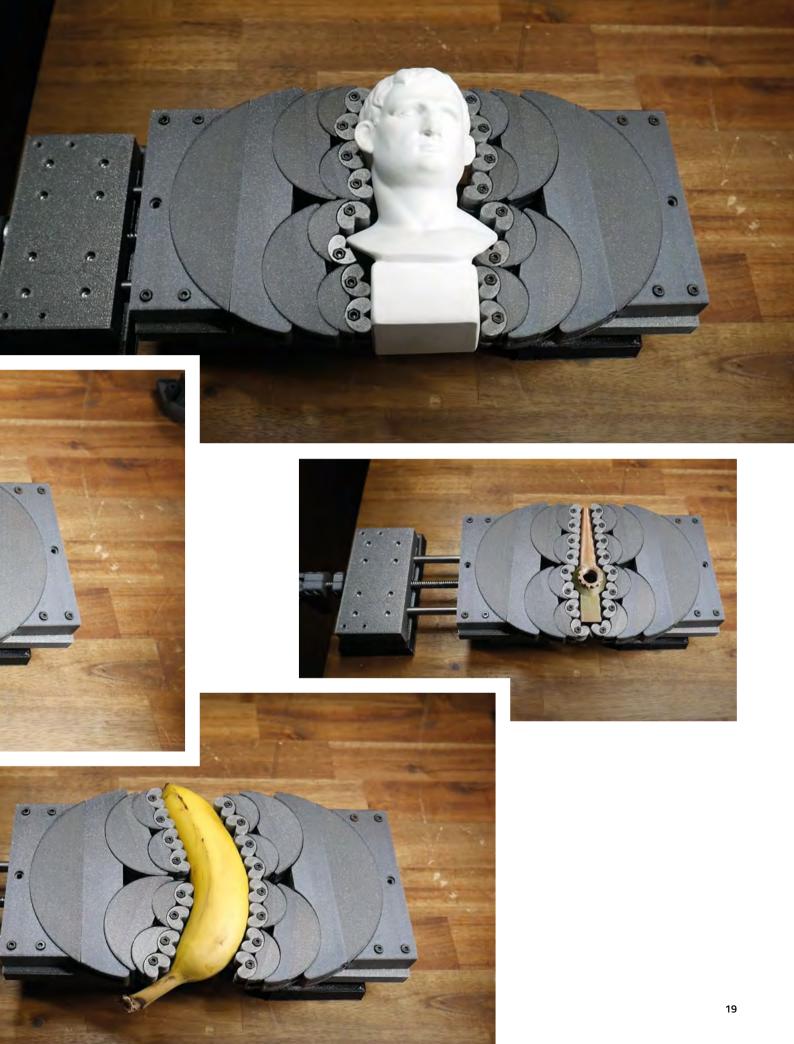
We're used to holding flat objects in a vice when we unleash our saw/chisels on them. We can even tolerate some slight wonkiness in those objects, as we've lined the jaws of our woodworking vice with soft pine, which deforms slightly so that the object we're holding doesn't have to. But we've never yet found a satisfactory way to hold a banana in a vice, or a bust of Julius Caesar – until now!

This fractal vice, designed and printed by Michael Laws of the Teaching Tech YouTube channel, uses a series of approximately half circles, stacked in layers that get smaller the closer you are to the gripping surface. It's surprisingly hard to explain – it's called a fractal vice because there's some maths involved that's too complex for our tiny brains – but the photos, and Michael's video, explain it perfectly.

➚ **hsmag.cc/FractalVice**

Meet The Maker:
Matthew Read

Clocks, Arduinos, and mechanical swans

Image Christopher Ison: West Dean College

Matthew Read is a conservator of the world's earliest computers (clocks) and the world's earliest robots (automata). He's worked at the Royal Observatory Greenwich (where time was invented); on several objects cared for by the National Trust, and the Silver Swan, an incredible 18th-century automaton housed in the Bowes Museum in the charming County Durham market town of Barnard Castle.

More than that, he's also an advocate for using modern technology, such as Arduinos in conservation, to take the strain off old, irreplaceable parts that are inevitably wearing away. We spoke to Matthew to find out what goes on when an expert fixes a broken clock, and learned a whole lot more than that.

"When you approach any broken object, I believe the important first step should always be the thinking rather than the doing.

> " Take it apart, photograph everything, measure everything, **look for damage or faults or wear as you go along and make notes** "

"A lot of students will emotionally and physically rush up to an object and get their nose in there with all the cogs and sprockets and whatever.

"And, inevitably, without sounding critical, they'll lose objectivity. I think one of the most important parts of [the process] is pushing [the object] as far away as you can emotionally, and then bringing it up really close. So

you see those nuts and bolts and manufacturers' marks and things.

"The first question is, where do we want to be at the end of this project? A lot of people would say, of course, we want to fix it, we want to put it back to how it was, we want to restore it – that totally meaningless word that we use the whole time.

"In terms of mechanical process, the first thing to do is to make the thing safe, so it doesn't fall over, take your fingers off, that kind of thing. So letting down or letting off the power is number one – the springs inside clocks can cause a lot of damage, so they have to be powered down. Then after that, you have a process: take it apart, photograph everything, measure everything, look for damage or faults or wear as you go along and make notes, and then wash the whole thing in mineral spirits. Then reassemble – broadly, the reverse process, and oil; put it back together again, and you're good to go. Then of course, somewhere in the mix you've got repairs, which usually entails making new components from scratch, as you can't just go out and buy them – I've had all sorts of iterations of workshops with machine tools: a lathe and the milling machine and all that kind of stuff.

WHEN IS A CLOCK NOT A CLOCK?

"Let's say, for example, that a clock ticks, it keeps time, strikes the hours on the hour; most would say that they want to restore those functions. So that means having the thing running within a reasonable amount of safety to the object and the person, keeping time striking the hours, that kind of stuff, within reasonable tolerances.

"So then begins the mechanical process. What is the minimum intervention we need to do to get to
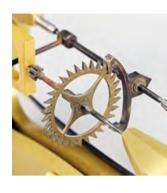
that state and nothing more, so no refinishing like you would on some of the restoration world.

"It's disappointing for most people because there's this kind of idea that you present the thing 'restored' and it's OK. And of course, in the case of the Silver Swan, it's never going to be OK – it's 250 years old; it's designed to break. It's been through many different hands; the threads are all worn. I'm just one tiny bit of some ongoing, fluid process. I respect *The Repair Shop* on the BBC for bringing restoration to a whole wider audience. But they kind of present it like there's a beginning here, there's a history, and then we present it. Ta-da! There's none of that in my practice.

"The whole conservation is built on cost/benefit. Not the [monetary] cost, but the wider cost of change to the object, and the benefit you're going to get from that. The risk – there's always risk in change. And the fact is that all things – me, you, the computer with the camera, and the building – are going into the North Sea, whether we like it or not. Somehow we are negotiating with that entropy.

"Since the 1980s, conservators have said that we're looking after this thing for future generations. And we realise now that this is sort of conceited; it doesn't work, because futures are fluid. We can't speak for those people in the future. So there's much more of a focus on the here and now, which of course is difficult. Because ultimately, when you work, when you're doing your job, your vocation, it's not actually about fixing clocks – it becomes about you. Then you get into existential questions. And they're the really tricky things, which I think is why people either like to concentrate on the made-up past or the fluid future, because it's a damn sight easier than the here and now.

"There's a great book called *The Wheelwright's Shop*, by [an author] called George Sturt. I think it's from the 19th century, somebody who was an academic, they go to a wheelwright's shop to see this process of learning, and they realise that it's absolute hell: nothing's written down, terrible conditions, grumpy people, lots of abuse, and the whole system of learning was really grim. And of course, it's all →
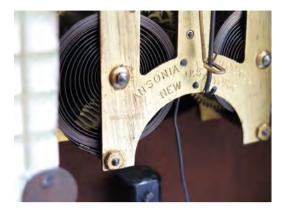
Below ◆
The aim in restoration is to change as little as possible, but sometimes you have to make new components in order to restore function

romanticised nowadays, the whole apprenticeship thing. My co-author John and I realised that there actually isn't a book for beginners in clock repair, even though there are lots of books that say on the front 'This is the book for beginners in clocks'. But actually, the minute you open them, you realise that they're basically people showing off, or there's a lot of assumed knowledge, or there's a lot of language that nobody knows what the heck it means.

"We thought we would actually write a book that helps people get off the starting line, which does have to be quite prescriptive: 'remove Part B, remove Part C, put it in G, and then you get your clock ticking', that kind of learning by rote thing. But once you've done that, you've built confidence.

"In the institutional world of museums and things, a lot of the aim of the system is to defeat people. It's not actually to encourage people to get involved. So we decided we'd write that book. And, the next book, one that we're doing at the moment, is actually that

next step: hopefully you've got some confidence, you've got a clock ticking, but actually, that is the easy bit.
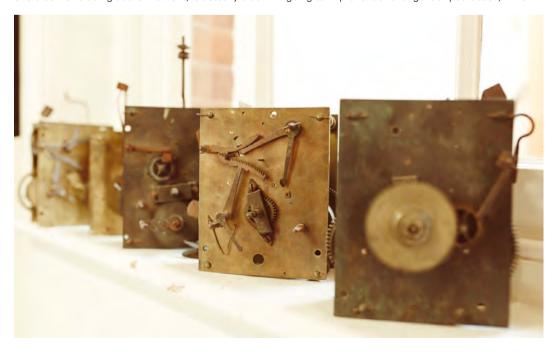
"Don't run your practice by listening to other people, by believing anything anybody says, because your practice is about you. And you're developing your attitude and process, which is why on popular programmes and magazine articles where people say, yeah, this is the way to do it – this is the proper way. There is no proper way, unfortunately.

"Objectivity is really important. I think one of the foundations of conservation is that you don't try and improve.

"The kind of people who get involved with clock repair… many of them are second-generation engineers or something like that. Really highly trained people, inevitably, they see an old thing, and they think they can make it better.

"And I say, 'No, no, no, don't make it better'. [There's] a really simple answer to this, and it's the relationship between craft and new making. If you're powered to do something really well, just make a new thing. Get a Raspberry Pi, get some brass, get some cogs, buy some robotics, whatever, and make a new clock.

"Again, it's like that thing of the here and now: it's much more difficult to do that than it is to muck up somebody else's work. And of course, the irony of this stuff is that the minute you say that's wrong, I'm going to improve it or change it or put it back, which

you can't do. I bet you a pound to a penny the next day something comes along and makes you look like an idiot. I had a really good example of this with an 18th-century clock dial – a grandfather clock dial. They're engraved, and the engraving is filled with wax, which is normally black to make the numbers stand out. This person who was teaching me said, 'Oh, they're never filled with red wax – you should take that red out and replace it with black', which I did, [because] I was a student. Literally the next day, somebody walked in with an 18th-century clock with the same red wax.

"It's too dangerous to impose that much on old stuff, because what you get is a normalisation. The minute somebody says, 'all steam-engines are green', you're stuffed really, because you get a normalisation. You get loads of 18th-century clocks that essentially all look the same. They've been normalised, and we know that life isn't like that.

ARE ALL REPAIRS EQUAL?

"There's no such thing as a bad repair or a good repair – as far as I'm concerned, those things don't exist. What I would say is that, where there's a public safety or damage to property issue, then, of course, you might want to look at that and say, well, this clock weight is gonna fall out of the church tower and kill somebody, then obviously you have to intervene. But if it's something like a bit of welding or brazing that looks ugly – who says it ugly? Because one person's ugly is another person's beautiful, and the minute you start normalising things, it never ends. Because then you get a subdivision of the good, and then say, actually, yeah, we don't really like that kind of thing. Because stick to this, and it never ends. It's a slippery slope.

"If you want, say, a tune player to play again, look at the options: you can stick a Raspberry Pi in there, and you can record the music digitally, which is what we did with the National Trust pagoda clock. You can just *look* at the old mechanism; leave it as it is. Clean off the dust and the spider-webs, and move more slowly. What's the great rush and imperative to have things finished, when nothing is ever finished?
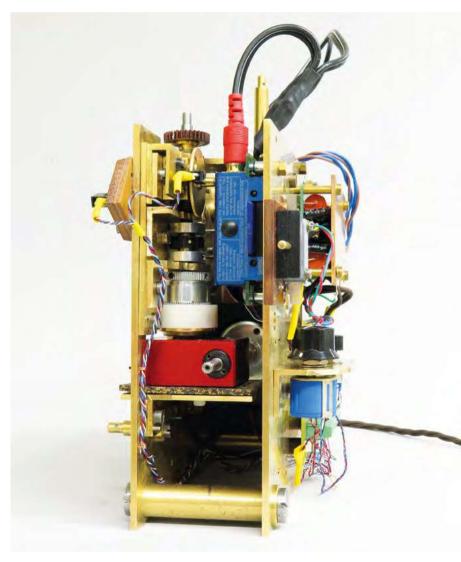
"I worked on the pagoda clock about ten years ago. And it was right on the borderline of where we could use discrete electronics, or use a microcontroller. And we went with discrete electronics in that case, but today, I would definitely use a microcontroller.

"The National Trust was really brave there. It was a postgraduate student project, so it was just me and a master's student working on the object, and the 18th-century automaton clock.

"We did the kind of traditional take it apart, clean it, photograph it, put it back together. But we found that one part of the module – the 18th-century stuff is modular, like new stuff is – was really worn. It had a lot of alterations in the 1930s. And yeah, we could make a whole lot of new parts, but actually, the Trust decided that it would create a new electro-mechanical module, which was sent over to me. We made it reversible (as in, we matched all the 18th-century screw threads and things), and we fitted it into the case so that it couldn't be seen. It was incredibly groundbreaking at the time, but of course, it raises the question…

"When I used to work at the Royal Observatory, with clocks by John Harrison, really famous sea-going clocks, people would come from all over the →

This is an example of a Fusee mechanism, as used in most English clocks made between 1650–1850

world, and the first thing they would ask is, 'are these the real thing?' The minute you say yes, they're happy – they've made the pilgrimage. But the minute you begin with microcontrollers and electromechanical stuff, you have to look at a different kind of picture.

"I think something like the Silver Swan is absolutely crying out for that kind of intervention. It was built in the 1770s – it's 250 years old or something. And the mechanics were never meant to be seen – you're meant to see a swan that looks lifelike.

"It plays six tunes now (it used to play eight on twelve bells). You could record that you 'retired' the original mechanism, and you can play it digitally. Nobody can tell, so long as you are honest, and you say this is digital music that you're hearing. But the really exciting thing about it is that, once you record it, of course, you can do what you want with it; you can make it play *Happy Birthday*. Or, more importantly, you can get people involved in education; you can put the Swan's tunes on SoundCloud, for instance.

"But anyway, we took it all apart – that was the first time since the 1970s – photographed it, because they washed it, put it back together, oiled it. And then what you should do with good practice every year with working objects is basically to have a peek at that oil and check it hasn't become contaminated with dust from the air, and so on. Twelve years later – it had been over a decade since we did the major

project – the museum quite rightly said, this is getting to the point where we obviously have to consider doing that again. And then Covid struck.

"We agreed that we wouldn't run it daily, and instead we'd make a film. So I popped along, spent a few days doing some strategic billing, fixing a thing or two. And then we wound it for filming. And we generated the film, which of course is great, because, you know, within reason, only a few people can get to Barnard Castle, but everybody can get online to see the film. So I think again, that's a bit of democratisation.

"I strongly encourage conservators to do their work in public, which a lot of them understandably are reluctant to do, because it's like a whole new scary learning process.

"When you look at other automata, you kind of look at them and think, well, yeah, that's clever. There's a lot of levers, cogs, and sprockets, but there's no way that it's lifelike. Whereas when you look at the swan, even to me, I've seen it hundreds of times working, and it's still crazy realistic, and it makes the hairs stand up on the back of your neck. It was taken from real life – they had dead swans on the slab to inform the work of the silversmiths.

"A couple of years ago, I saw an X-ray of a swan that had been shot with a shotgun. There were 24 vertebrae in the swan's neck. I said, 'wait a minute', and consulted my notes; the automaton has 24 vertebrae – they'd copied life that closely. Which is

> **Only a few people can get to Barnard Castle, but everybody can get online to see the film**

why it looks like it's the world's finest automaton of its period, because that's what the maker was trying to do to simulate life.

"There's something called a Fusee-driven movement, and it's a mainstay of English clock-making from the middle of the 17th century until the middle of the 19th century. There are three of these inside the swan: one drives the water, one drives the music, and one drives the swan's neck. The swan differs from all other automata in one respect, and that's because the one that drives the neck is kind of like a racing car, where the engine and the gearbox form part of the structure of the whole – it's called a a monocoque in cars.

"It's crazy different from other automata where the maker got a clockwork motor and cut bits off and stuck bits on and used it to power their automaton. The Waddesdon elephant is a little bit like that.

FROM AUTOMATA TO COMPUTER

"So anyway, there are three clockwork motors. The main one has a single cam that has multiple cams on it; it's also got five chains that run up the neck which makes the neck move.

"So it's a single cam body with, I don't know, ten different cams on it that do all the different things.

"Speaking of computing, Charles Babbage bought one of John Joseph Merlin's automatons [the maker of the Silver Swan]. When I went to the Science Museum with my students to look at clocks, I was looking at the Difference Engine and realised, while looking at the cams, that they were just like the cams on the Silver Swan. Babbage probably got his inspiration for those rotating, sliding, twisting cams from an automaton like this one.

"There are rules to conservation; one of them is, anything you do should be reversible. So if you're fixing an electric motor into a thing, for instance, should be able to undo that. The big challenge there is that before the 1850s, the threads on screws were not standardised, so you can't go down to B&Q and buy a nut and bolt that will fit.

"So you've got to make all the threads, which is not impossible, it just takes quite a bit of time. Obviously, you want to be as transparent as possible about any changes you make. So, I think what the National Trust did was, they took the module that we had taken out and just had it at the side with a text panel putting it in context.

"There was a lot of resistance at first: the staff there were convinced that they would be able to tell that we'd changed something. When I installed the module and played the digital music, somebody walked into the room and said, 'It's so great to have

our clock back! But come on Matthew, let's hear this terrible jingle-jangling digital stuff'.

"A friend of mine is the country's foremost turret clockmaker. So he services 1000s of turret clocks around the country. And he fits them all, reversibly again, with sensors and a microcontroller that sends him a health report every single week. He deals with Salisbury Cathedral, Canterbury Cathedral, you name it, and they send him an email every week with a health check. So if you've got a problem with lubrication, which is a big deal with a clock, and it starts needing more torque to run, or there's a fault with it striking, you don't have to wait until it becomes a disastrous failure. You can ring up the museum and say 'stop that object – we're going to pop around and have a look and see what the problem is'. That's not to say that everything is suitable for adding microcontrollers and sensors, but there's a place for it. Let's move things forward". □

Below ◆
The Silver Swan at Bowes Museum was designed and built in 1773 by John Joseph Merlin

Image
Claire Collinson Photography: The Bowes Museum

New tools, old tricks

Printing a new–old house, one piece at a time

Samantha Snyder

🐦 @SamanthaSnyder

Sam is one of Raspberry Pi's representatives in our former North American colonies; when she's not doing that, she's restoring an old house brick by brick, fuelled by coffee and PLA.

Humans are natural tinkerers. When we get our hands on new tech, our first instinct is to push the limits of innovation and see what novel (and often useless) creations we can make. But sometimes the best part of new tools is the ease in which they can accomplish well-known tasks. If you've strolled down the lumber aisle of your local home improvement store, you'll know that the price of lumber has skyrocketed during the pandemic. As someone who is remodelling an old brick Victorian-style home, the thought of possibly wasting expensive wood on custom millwork prototypes makes me question my need to over-engineer even the simplest of projects. Instead of rethinking my life choices, I turned to my favourite cost-saving obsession – 3D printing.

While I enjoy testing prints and techniques that push the bounds of my machines, there's a certain level of satisfaction in how well 3D printers can simply…make things. Especially familiar things. Whether I just want an uncomplicated yet unique rosette, a structurally sound corbel, or a complex ceiling medallion, it's faster and cheaper to iterate using a 3D printer – not to mention less of a mess. Once I have the final design haphazardly balanced in place, I can decide to have it milled from wood, or be lazy and just paint and secure the print.

3D printing isn't always the best solution to every problem, and that's OK. Trying to print a replacement part that is inexpensive and widely available can be fun, but generally takes more time and effort than it's worth. 3D printers have always shone brightest when rapid prototyping is needed – which is just a fancy way of saying the first attempts are expected to fail. A lot. And when I attempt home improvement projects for the first time, I tend to fail. A lot. Using my 3D printers makes those failures a little less painful. Is it a traditional solution that my local historical society would likely approve of? Probably not. Does it accomplish the goal of letting me add personal touches and hidden geeky Easter eggs to my house, without feeling like I need to invest in lumber stocks? Unfortunately for the next owner, yes, yes it does. ◻

> There's a certain level of satisfaction in how well 3D printers can simply… make things. Especially familiar things

Letters

GIGA PRINTER

I've never been a big fan of the phrase 'nobody wants to see how the sausage gets made', mostly because I very much do! Seeing the unglamorous bits that go into a creative process is fascinating, and it makes the final product even more impressive. So the best bit of your chat with Willow Creative last issue wasn't the costumes that she makes – it's the 3D printer hack. Anyone can order a few bits of extruded aluminium online and swap them for other bits of extruded aluminium with the same profile, meaning you can make the things you own bigger or smaller to suit your needs. It's a simple concept, but seeing a concrete example really helped me get my head around it.

David
Bury

Ben says: We've said it before, and we'll say it again: open-source hardware puts the power back into your hands. There are doubtless manufacturers who would prefer you 'upgraded' and bought a bigger printer when your existing one isn't big enough, but the ability to modify your machinery is a big draw for us, and it is the reason we'll choose open-source time and time again.

CONCRETE PRINTING

I saw a 3D-printed house in your magazine a while ago, but did you know that there's a giant 3D printer being used in England to build part of our new high-speed rail line? At first, I thought destroying thousands of trees in order to get from London to Birmingham 20 minutes faster was a poor bargain, but now I'm far more enthusiastic about the whole thing.

William Mackenzie
High Wycombe

Ben says: As I understand it, they're using 3D printers to speed up the construction of some of the infrastructure around the rail line. At the end of the day, it's a tool, and one that, when used properly, can fill a niche. We're more excited about the possibilities of printing in chocolate right now.

WEARABLES

Please pass on to Kitty Yeung [Crowdfunding Now, HS45] that she is to cease making women's wear forthwith, and switch to men's jackets. The work she's doing is absolutely brilliant, but as a 41-year-old chap, I just don't have the figure to carry off a fitted dress.

Paul
Middlesbrough

Ben says: I feel the same way whenever I'm shopping for kids' clothes. Why can't you get dinosaur pyjamas in adult sizes? We'll forward your constructive feedback to Kitty and see what she says, but in the meantime, why not see what you can conjure for yourself with a bit of magic from our collection of wearable technology projects: hackspace.raspberrypi.org/books/wearable.

CROWDFUNDING
NOW

DiamondBack 3D printing nozzle

The toughest accessory?

From $76 | kickstarter.com | **Delivery:** November 2021

Diamonds are forever, or so the De Beers marketing team would like you to believe. While not technically indestructible, they are very hard and wear-resistant and also have a few other useful properties. They have very low friction and very high thermal conductivity (actually, they have the highest thermal conductivity of any known material, and this is commonly used by jewellers to distinguish between real diamonds and imitations). All this means that diamond is a great material for 3D printing nozzle tips.

Obviously, the big question is the price. At $76, it's not cheap. You could get over 100 brass nozzles for this if you shopped around (they wouldn't be good nozzles, but they would be nozzles). What you're getting for your money is lack of hassle. Since this is one nozzle that you should just be able to put in and forget about, you potentially save a lot of time. How much time depends a lot on what you

3D-print. If you're printing a lot of abrasive material, then even hardened steel nozzles can wear out. Ruby is tougher, though not as tough or thermally conductive as diamond, and similarly priced. You're

> **If you're printing a lot of abrasive material,** then even hardened steel nozzles can wear out

also buying one less thing to worry about. When print quality starts to slip (as it tends to with 3D printers over time), you can be confident that the problem isn't a worn nozzle.

If you're printing mostly non-abrasive material, then you'll see less change as other nozzle materials won't wear out as fast. ◻

PRECISION PRINT

ANY FILAMENT

LONGER LIFE
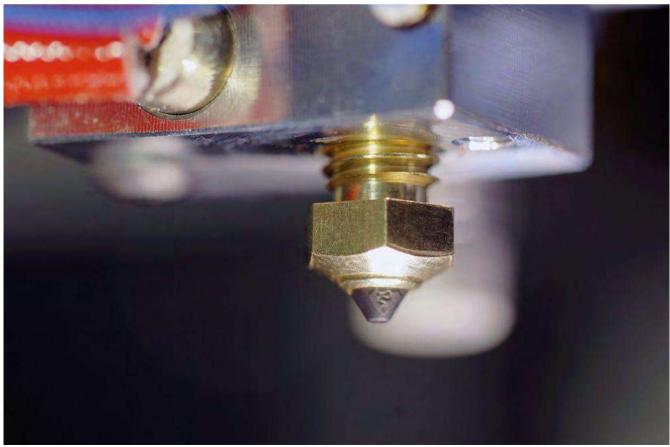
BUYER BEWARE ⚠
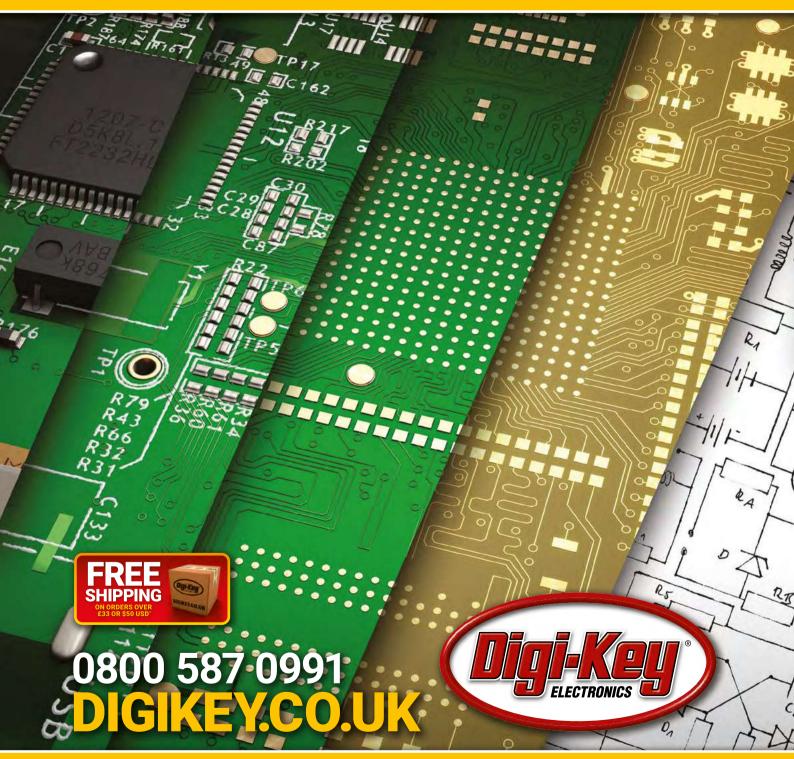
When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

GET INVOLVED IN
CITIZEN SCIENCE

COLLECT YOUR OWN DATA AND SHARE IT TO CONTRIBUTE TO SCIENTIFIC RESEARCH. THERE'S A CITIZEN SCIENCE PROJECT FOR EVERYONE

BY PHIL KING

n fields ranging from astronomy to zoology, there are thousands of citizen science projects running worldwide. Many of them can be found listed on **zooniverse.org**, helpfully filtered into categories. While some projects are limited to a particular locality, others are global in nature.

Many projects don't even require any equipment to take part, such as the RSPB's annual Big Garden Birdwatch, which asks participants to spend an hour watching their garden and manually noting down numbers of different bird species on a printed chart.

Others require specialist kit, such as one of the first-ever citizen science projects, Operation Moonwatch. Established in 1956, at the dawn of the space age, it relied on thousands of people worldwide using specially designed telescopes to monitor the night sky and record sightings of artificial satellites such as the original Sputnik 1. With the data obtained, scientists were able to track their trajectories.

What all citizen science projects have in common is the participation of ordinary citizens recording data that may well be of use to the scientific world. Only last year, the network of home-based Raspberry Shake earthquake-monitoring devices provided valuable data used in a scientific paper by seismologists detailing the decrease of human-related seismic activity during pandemic lockdowns.

Space-themed projects are popular. Harking back to Operation Moonwatch, the Global Meteor Network comprises volunteers using DIY skyward-pointing video cameras to monitor meteors in order to calculate their orbits and help to reveal the presence of larger 'parent bodies' such as comets.

Environmental monitoring projects are also plentiful. Sonic Kayaks are fitted with a range of sensors to monitor the underwater climate, often in difficult-to-reach places. A Smart Buoy can take wave measurements to study the impact that sea levels are having on coastlines. Meanwhile, the Sensor.Community monitors air quality worldwide using data from homemade stations – we'll show you how to build one and start uploading your data.

Not only is citizen science a great way to crowdsource the collection of valuable scientific data, but it's also typically educational and a lot of fun. Why not join in? →

SENSOR. COMMUNITY

sensor.community

You'll Need

- NodeMCU V3
- Particulate matter sensor (e.g. SDS011, PMS5003)
- BME280
- Jumper wires
- PM pin breakout if using PMS sensor
- 2 × plastic drainpipe bends

Formerly known as Luftdaten, in five years this superb citizen science project has ballooned from a local air quality monitoring effort in Stuttgart to a global network of over 14,000 sensors in over 70 countries. These sensors collect valuable data on particulate matter levels, along with meteorological conditions. The open data generated is available for use by scientists and institutions.

BUILD AN AIR QUALITY STATION

To contribute data to Sensor.Community, a kit is available with the software pre-flashed onto a NodeMCU V3 board. However, it's fairly easy to build your own air quality station from individual components.

While the project supports a fairly wide range of sensors, the default setup features a BME280 sensor for measuring temperature, pressure, and humidity, and an SDS011 dust sensor for measuring levels of particulate matter in the air. Since we already had a Plantower PMS5003 to hand, we opted to use that instead of the SDS011.

Below
Data from thousands of air quality stations can be viewed on the online map

STEP 1
FLASH THE FIRMWARE
Connect your NodeMCU V3 to a computer via USB; the serial connection should work by default on a Linux machine, but you'll need to install the relevant drivers for Windows or macOS (see **hsmag.cc/SensorCommunity** for links). Make sure you get the correct one for your NodeMCU's chip type – ours was CH341.

Next, download the airRohr firmware flashing too from **hsmag.cc/airRohr**, choosing the relevant version for your computer OS, and install it. Run the application, choose **latest_en.bin** (or another language version) from the firmware version drop-down, and press Upload to flash it to the connected NodeMCU. Note: If the flashing application says 'No boards found', choose the option with 'usbserial' in it from the board drop-down.
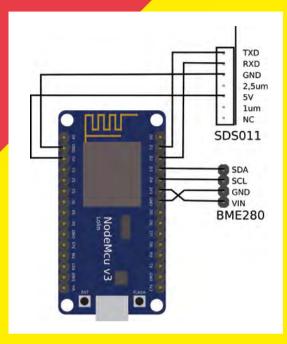
STEP 2
WIRE UP THE SENSORS
Connect your particulate matter sensor to the NodeMCU V3. The wiring for SDS011 and BME280 sensors is shown in **Figure 1**. For our PMS5003, we used an adapter (**hsmag.cc/PMBreakout**) to break out its PicoBlade cable to six pins and wired 5V to VU on the NodeMCU, GND to GND, TX (marked RX on the breakout) to D1, and RX (marked TX on the breakout) to D2. If your BME280 doesn't already have pins, you'll need to solder them on. You can find the relevant connections for other supported sensors at **hsmag.cc/SCSensors**.

STEP 3
CONFIGURE THE STATION
When first powered up, the air quality station will create a wireless hotspot with the name Particulate Matter ID, Feinstaubsensor-ID, or airRohr-ID. The ID number is the chip ID (for example, 13597771). Note this down, as you'll need it for registering the station.

Connect your computer to this wireless hotspot and point a web browser to **http://192.168.4.1** to access the station's web dashboard. From here, choose Configuration > WiFi Settings, select your wireless router, and enter the password for it.

Upon pressing **RETURN**, the station will restart and connect to your wireless network. Once it does, it will no longer be reachable at 192.168.4.1, but at its allocated IP address – it'll show up as 'Airrohr-' followed by its chip ID in your router's devices list.

Visit the station's IP address to continue configuration. In the Configuration > Sensors tab, make sure the sensors you're using are ticked. Hit 'Save configuration and restart' to save the settings.

After about two minutes, the device will now attempt to upload data to Sensor.Community, as shown in the 'Debug level' menu. But first you'll need to register it. Visit **devices.sensor.community**, sign up for an account, and go to My sensors > Register new sensor. For Sensor ID, enter the chip ID number you noted down; select 'esp8266' for Sensor Board. Fill in the Basic and Additional Information, then check that the correct sensors are selected in Hardware Configuration. Finally, select your station's location on the map and press 'Save settings'.

It'll take a couple of minutes for your station to start uploading data to Sensor.Community; you can check by clicking the Data button for it on the 'My sensors' page, and also 'show it on map'.

STEP 4
INSERT INTO PIPE
Unless you are using your air quality station indoors, you'll need to protect it from the elements. For this, a couple of

interlocking plastic drainpipe bends work well, with duct tape wrapped around the join – for better waterproofing, you could use silicone sealant.

The assembly guide on the Sensor.Community site advises using a cable tie to bundle the SDS011 dust sensor and NodeMCU together, with the latter's antenna facing away from the sensor to avoid interference.

It also instructs you to attach a flexible plastic tube to the SDS011 dust sensor's input to direct air from the end of the drainpipe to it. This wasn't required for our PMS5003 sensor, which we placed near the end of the pipe, attaching it with a sticky Command Strip. We also positioned our BME280 sensor near the pipe end, with the NodeMCU located further up.

To protect your sensor from invasion by insects, you could place a fine mesh over the pipe ends.

STEP 5
PLACE IT OUTDOORS
The final step is to place your air quality monitor outside for accurate temperature and air quality monitoring. We used long cable ties (link two together for extra length if needed) to position our enclosure on the underside of some guttering, but you could secure it to a downpipe with diagonally crossed cable ties.

To power our station 24/7, we ran a long USB power cable through a small window to reach our indoor mains socket; if you have a weatherproof outdoor socket, that's even better.

If the air quality station is a fair distance from your wireless router and signal strength/quality is low, consider placing a WiFi range extender nearer the station.

Congratulations: your air quality station is now ready to start sending valuable data to Sensor.Community. →

Figure 1 ◆
Schematic for wiring an SDS011 dust sensor and BME280 environmental sensor to the NodeMCU V3 (custom Fritzing part by Roman Minyaylov)

Below ◩
Place your pipe-housed air quality station in a suitable spot outdoors, preferably 1.5 to 3.5 metres above the ground and well-ventilated

RASPBERRY SHAKE & BOOM

raspberryshake.org

Above ◈
The standard RS1D Shake in its enclosure with a Raspberry Pi

Did you feel the earth move under your feet? It might have been a tremor. To investigate further, you can measure seismic activity with a Raspberry Shake device. Based around a Raspberry Pi and available in several variations (from $215), the Shake is a compact, professional-grade seismograph.

When placed on a flat surface, it uses a 4.5Hz geophone to monitor underground vibrations – even those too small to be perceived by human senses. Operating at 100 samples per second, the geophone converts ground movement into voltage; this analogue signal is then converted into digital data by the Raspberry Shake circuit board to be stored on Raspberry Pi.

This being a citizen science project, the high-quality data gathered is also sent to a central hub, ShakeNet: **shakenet.raspberryshake.org**. Since the initial Shake Kickstarter campaign back in July 2016, this community of 'Shakers' has grown into the largest earthquake monitoring network in the world, with the little seismographs making it to every continent.

As well as viewing your own data on the ShakeNet dashboard, you can see data from around 1500 other sensors worldwide on the Station View map. Other view options reveal earthquake activity, raw data graphs, a P and S wave picker to locate earthquake epicentres, earthquake audio files, and Boom infrasound sensors (more on that later).

Interestingly, some of the vibrations detected by Raspberry Shake are related to human activity, and there was a noticeable fall – up to 50% in urban areas – detected during the numerous lockdowns of 2020. An academic research paper on this phenomenon, publishing in Science magazine (**hsmag.cc/SeismicQuieting**), was based mainly on data from the Shake network.
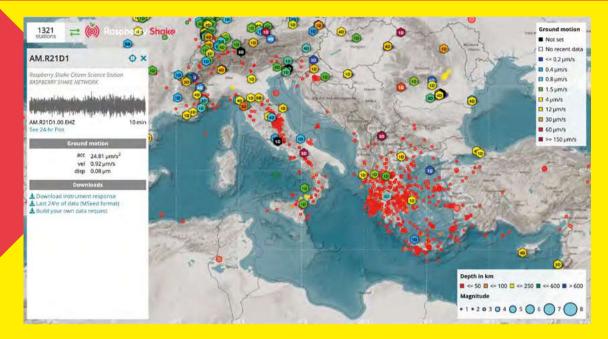
Raspberry Shakes also make excellent educational tools. Not only can they be used to teach all aspects of STEM, but they are also a fantastic way of raising earthquake awareness, especially in communities located in active earthquake areas.

MODEL SELECTION

The Raspberry Shake can be bought as a ready-to-use turnkey solution or a DIY kit. There are three main models:

- **RS1D:** The most affordable model comprises a vertical geophone, Shake circuit board, 8GB pre-flashed microSD card, and an enclosure.

- **RS3D:** For improved detection, the RS3D features three geophones (orthogonal east, north, and vertical) to obtain the most accurate ground motion data across all axes – this model is usually chosen
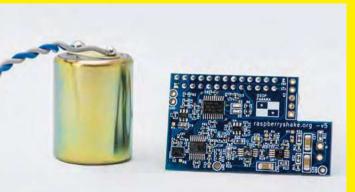
by professionals or citizen seismologists with a real passion for seismology.

• **RS4D:** This model was created specifically for active earthquake areas. In conjunction with the geophone, it features three MEMs accelerometers that measure local ground motion across all axes and can remain on scale even during strong shaking, which can sometimes saturate the geophone sensor.

For a detailed comparison between the models, go to **shop.raspberryshake.org/compare**. Alternatively, you can buy just the Shake circuit board and geophone and make your own enclosure. A step-by-step guide to setting up a DIY Raspberry Shake with these can be found in The MagPi magazine, issue 91 (**magpi.cc/91**).

Whichever model/option you choose, it's best to locate your Raspberry Shake in a quiet corner where it's unlikely to be disturbed. For best results, the makers advise placing it on a bare floor (no carpet): "A good location for the Shake would likely be on the concrete slab of the lowest floor, near a foundation wall and away from furnaces, washing-machines, air conditioners, and such." With a special all-weather (and rain-, dust-, and bug proof) enclosure, you could even place your Shake station outdoors.

Wherever you position it, it's recommended to connect the Shake's Raspberry Pi to your router with an Ethernet cable instead of WiFi, as the latter can interfere with the sensitive geophone's readings.

BOOM BOOM

While the Raspberry Shake measures and monitors the subtle seismic vibrations of the Earth, the Raspberry Boom was designed to record infrasonic waves: imperceptible 'noises' that bounce around the atmosphere. Although inaudible, these sound waves are ever-present and emitted from a wide variety of sources, both natural and human-related.

Using a sophisticated acoustic pressure transducer, the Boom is able to detect infrasonic signals from such natural sources as volcano activity, bolides (meteors that explode in the atmosphere), avalanches, landslides, tornadoes, lightning, and severe storms. Human-related sources include traffic, planes, gunfire, and nuclear testing.

Like the Shake, the Boom can be easily configured via a web interface and the data is viewable on a map: **boomview.raspberryshake.org**.

For the ultimate sensory station, you can purchase a combined Raspberry Shake and Boom in one enclosure for a professional all-in-one seismograph and infrasound monitor. These signals are often emitted simultaneously from sources such as rocket launches, volcanic eruptions, avalanches, and landslides. →

Above �
All Shake stations can be viewed on a world map. Click on one to see its latest ground motion data

Below ◹
Vibrations are detected by the sensitive geophone sensor, and the signals converted into digital data by the Shake circuit board

SONIC KAYAK

hsmag.cc/SonicKayak

P art environmental sensing system, part musical instrument, the Sonic Kayak has been developed over the past five years by the team at Then Try This (formerly FoAM Kernow), led by Dave Griffiths and Dr Amber Griffiths.

Kayaks rigged with underwater sensors gather valuable environmental data while also generating live music based on it, providing the paddler with an extra dimension of senses with which to explore the underwater climate.

"This type of data is really hard to get, particularly for places close to cliffs or in tidal estuaries, where it's difficult to take large research boats," says Amber. "Our system also means you can get very fine-scale data, as opposed to the broader scale sea temperature data you get from satellites, or the data you can get from attaching sensors to buoys."

Each Sonic Kayak kit features four environmental sensors. Water temperature is measured using a digital thermometer, water cloudiness using a homemade

sensor consisting of an LED and an LDR in a dark tube (the amount of light that hits the LDR is a measure of how cloudy the water is), underwater sound using a hydrophone, and above-water air pollution using a laser dust sensor. There's also a GPS unit to record the kayak's geographical co-ordinates, which enables mapping of the data gathered.

The kit's electronics – including a Raspberry Pi 3B and Arduino Nano – are housed in a BOPLA Bocube plastic enclosure that fits into the rear of the kayak. To prevent damage from any contact with water, the electronics are embedded in resin, while rubber seals are used for the sensor cables.

Then Try This is interested in getting the kit into the hands of people who want to use it to map their local environments, and have recently been doing remote training for a sound researcher to build her own. If you fancy making a Sonic Kayak kit, the bill of materials, instructions, and software are available on the wiki (**hsmag.cc/SonicKayakGitHub**).

As of yet, there's no system for people to submit data, although the team hope to eventually set up an online map portal to visualise it. For the time being, data is being uploaded to Zenodo (**hsmag.cc/SonicKayakData**) for anyone who wants to check it out.

Right →
Connected to various environmental sensors, the electronics are housed in a BOPLA Bocube plastic enclosure that fits into the rear of the kayak
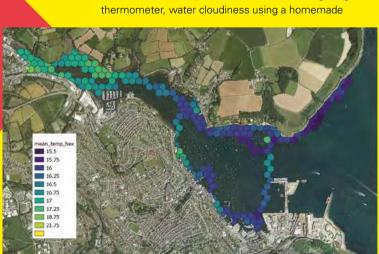
Below →
One of the first attempts at mapping the underwater temperature data. Some uploaded data can be viewed at hsmag.cc/SonicKayakData

mean_temp_hex
- 15.5
- 15.75
- 16
- 16.25
- 16.5
- 16.75
- 17
- 17.25
- 18.75
- 21.75

WECOUNT

In order to learn more about traffic flow and create better routes to avoid congestion, city planners need to know how many vehicles are being driven past a particular point, what type they are, and how fast they're going.

While one rather laborious way to gather such data is to have people stand on the pavement and count vehicles manually, some governments have used expensive technology to monitor traffic, often limited to a narrow set of locations such as highways. The WeCount citizen science project offers an affordable alternative to gather data from a wider range of locations using inexpensive traffic sensor units powered by Raspberry Pi.

Currently operating in six European cities – Madrid, Barcelona, Ljubljana, Leuven, Dublin, and Cardiff – WeCount is in the process of expanding to cover other towns. The project invites members of the public to place a Telraam traffic sensor unit in their homes, looking out of a window at the street to count vehicles. In some locations, an air quality sensor may be added to monitor pollution.

The Telraam units are powered by a Raspberry Pi 3A+ with a standard Camera Module capturing video at around 30 fps, to catch even the fastest cars over a few frames. Standard Python libraries are used along with OpenCV for the image processing and object tracking algorithms.

Simple background extraction and contour detection methods are employed to find moving objects – including cyclists and pedestrians – on each frame; an algorithm is then used to identify and track them over their visibility period, stretching across a set of consecutive frames.

Data is uploaded to the cloud so that it can be used within initiatives related to air pollution, safety, active travel, noise, and speed. Live traffic maps of neighbourhoods – filterable by cars, heavy vehicles, bicycles, and pedestrians – are also available on the Telraam website (**telraam.net**) and make for fascinating viewing. →

GLOBAL METEOR NETWORK

globalmeteornetwork.org

Above
An impressive multi-camera setup in Montgomery, Wales

Image
Roger Banks

Many space-related citizen science projects require participants to analyse collections of existing photos and other data such as radio bursts. Global Meteor Network takes a more hands-on approach to observing the night sky, although it makes use of cameras rather than telescopes.

Did you know that every day, hundreds of tons of small interplanetary objects enter Earth's atmosphere? These tiny, typically pea-sized fragments can offer us a unique view on the formation and evolution of our solar system. Yet most of them remain unobserved.

The Global Meteor Network's mission is to observe these meteors, aka shooting stars, using a global network of cameras pointed at the night sky. Each camera is connected to a Raspberry Pi running open-source software for video capture, compression, and meteor detection. Currently, the project has over 200 camera stations based in 20 countries, with data published daily on the project website.

To join the network, you can either buy a ready-to-use meteor camera or build your own from scratch (**hsmag.cc/BuildAMeteorCamera**) – the parts cost

around €200 (£170). An ultra-low-light camera and Raspberry Pi are placed in a waterproof enclosure attached to a heavy-duty mounting arm to secure it to a wall. Power and a network connection are supplied via a Power-over-Ethernet cable, injector, and module board. If you live in a cold climate, an optional heater can be added to the setup. Software-wise, a pre-built OS image is available for Raspberry Pi – a 128GB microSD card is recommended.

With the camera in place and pointed skywards, you then send its location, azimuth, and elevation to Global Meteor Network (so they know which part of sky it's covering) to generate a unique station ID. Detailed instructions for camera setup and calibration are given on the website.

Not only is it fascinating to observe meteors, but the project also has a serious purpose. With at least two of these systems whose cameras are looking at the same volume of the sky, meteor orbits can be calculated, revealing from which part of the solar system they emanated. Crucially, it can point us to where to look for unknown parent bodies – there could be a comet out there on a collision course with Earth!

SMART BUOY

You don't need to join an existing citizen science project to get involved in gathering valuable scientific data. You could always start your own project based on whatever field you want to study. That's what the small T3chFlicks team did with their Smart Buoy.
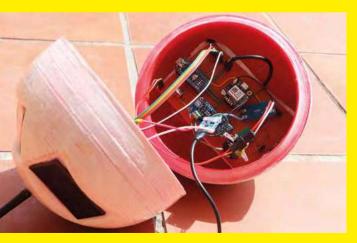
The project came about when team member Steph was writing her undergraduate dissertation on the impact of sea level rise in Grenada and realised how difficult it was to get reliable, continuous data on the impact that sea levels were having on coastlines.

"The coast is a dynamic area at the mercy of waves," the team note. "Rising sea levels nibble at beaches, and powerful extreme events like hurricanes completely decimate them… To understand how to save them, we need to understand the forces driving their change."

After six months of planning and designing, including "two jam-packed weeks" in which the team physically built and deployed their prototype, Smart Buoy was finally deployed off the coast of Grenada.

3D-printed in two halves, the Smart Buoy's case is waterproofed to protect the electronics inside. It is also equipped with four mini solar panels connected to a battery for power.

Below ◆
3D-printed in two halves, the waterproofed Buoy is fitted with four mini solar panels for power for the electronics crammed inside

Left ◆
Testing the Buoy indoors, before placing it out into the Caribbean Sea

Above ◆
Hosted by the base station's Raspberry Pi, an interactive web dashboard shows the data from the Buoy

Inside, an Arduino Nano is connected to a GPS module to log the Buoy's position, along with sensors to take wave height, wave power, and water temperature measurements. These are then sent by a radio module – with a maximum range of 1 km – to a Raspberry Pi base station onshore, which hosts a real-time dashboard (using Vue.js) and stores the measurements in a database.

While, as yet, there's no central hub for other users to submit their data, you could make your own Smart Buoy device using the detailed instructions and source code on the project's GitHub page (**hsmag.cc/SmartBuoy**).

The team would love to hear from anyone who could help them take the project to the next level. "It would be amazing to see the Buoy in production," says Steph, "and we believe it would be a really valuable tool for education, citizen scientists, and even small governments." □

FEATURE

How I Made

CRYSTAL RADIO

Turn the clock back by building a radio that needs
no battery or power supply

By **Mike Bedford**

You've probably never tried taking your smart speaker or DAB radio apart, and I don't advise it, but you can probably guess the sort of thing you'd find inside. Given the sophistication and complexity of most devices used to receive radio stations today, you might find it surprising that we're going to show you how to build a radio receiver and, what's more, one that isn't based on a digital platform like a Raspberry Pi. Instead, we're going to build the radio from scratch using the simplest of components. Perhaps even more amazing is that the radio doesn't need a battery or any other source of electrical power.

The type of radio we're featuring here is called a crystal radio. Designs like these go back to the earliest days of radio broadcasting – which takes us back to the early 1920s – yet they'll still work today, thanks to the continuation of AM radio stations in the medium wave band. With a crystal set, as they were called in the 1920s, you could only listen on a pair of headphones. Despite that, because early radios with speakers were expensive – typically £8, or almost £500 in today's terms with inflation taken into account – crystal radios were very popular. What's more, because they were so simple, those early listeners often built them at home, thereby saving even more money. So, building your own radio is quite unashamedly a retro exercise, turning the clock back by 100 years.

THE COMPONENTS
For the simplest of crystal radios, you need just five components: an inductor, a variable capacitor, a diode, a fixed value capacitor, and an earpiece or headphones. You'll make the inductor yourself, so that leaves just

four components, and you can probably manage without one of the others. Variable capacitors are rarely used today, so you'll probably buy yours as a 'new old' component – that's one that was made a long time ago but never used – on eBay. It's not ultra-critical, but the value you need is between about 350 and 500 pF (this is the maximum capacitance; the minimum might be about 5–10%). Ideally get a dual gang variable capacitor, because it'll allow you to try further circuit configurations later. These will often be shown, for example, as 2 × 350 pF, which would mean two 360 pF variable capacitors that are tuned together. You also need a plastic knob for the variable capacitor, because if you touch its spindle with your fingers, you'll alter its tuning. Next up is the diode and, although most diodes are silicon-based, those won't work in a crystal radio. You need a germanium diode, possible part numbers being OA90, OA91, and 1N34A, although there are others, and you can get them from ordinary electronic component suppliers. The headphones or earpiece must be high impedance, so this excludes nearly all modern headphones. High impedance earpieces are available on eBay. The fixed capacitor should have a value of about 1–10 nF, but this isn't critical; in fact, the earpiece might have sufficient internal capacitance that it isn't needed. Fixed capacitors can be bought from any component supplier.
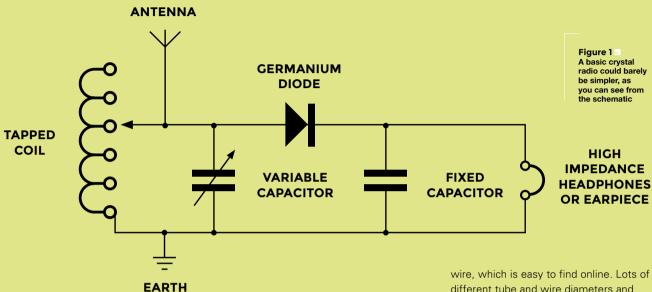
Now to the inductor, which you'll have to make yourself. An inductor is a wire coil, and you need to create this by winding the wire around an insulating tube. It's best to use a plastic tube and enamelled copper →

> You also need a plastic knob for the variable capacitor, because if you touch its spindle with your fingers, you'll alter its tuning

Above
This project takes us back to the 1920s when a crystal radio was the latest 'must-have'

Left
Winding the coil isn't difficult, but it does require some patience

ANTENNA

GERMANIUM DIODE

TAPPED COIL

VARIABLE CAPACITOR

FIXED CAPACITOR

HIGH IMPEDANCE HEADPHONES OR EARPIECE

EARTH

Figure 1 🗗
A basic crystal radio could barely be simpler, as you can see from the schematic

We suggest mounting the components on a wooden board. It's easy to figure out how to construct your crystal radio this way

Right 🗗
To tune the radio, you just need to tweek the variable capacitor

wire, which is easy to find online. Lots of different tube and wire diameters and numbers of turns have been used, but we suggest you follow our lead. We used a 55 mm diameter tube and 22 SWG (Standard Wire Gauge) wire, which is about 0.7 mm in diameter, and the closest American equivalent is 21 AWG. We used 100 turns, which is more than we really needed, and if you're using a different size of tube or wire, we suggest you also wind plenty of turns, because it's easier to use only part of the coil than to add more turns later. To be able to use only part of the coil, you need to add connections called taps after every ten turns or so. To start, drill two small holes near one end of the tube and thread the end of the wire through them to secure it in place while you wind the wire. Keep the turns close together, and make a twist wherever you want a tap. If you get bored – as you probably will – secure your work with insulating tape so it won't unravel before you return to it. When you're done, secure the other end by threading it through another pair of holes.

Diodes and Crystals

The name crystal radio takes us back to the early days when diodes were valves (called tubes in America) which needed power. But although there were no semiconductor diodes, a crystal of the mineral galena could be used as one with a piece of stiff wire called a cat's whisker.
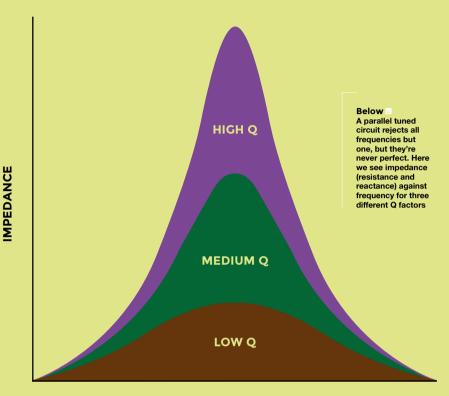
Finally, carefully use a very sharp knife (e.g. a craft knife) to scrape the enamel insulation off the wire at each of the taps.

WIRING IT TOGETHER

Now you need to wire the components together, following the schematic (**Figure 1**). Although diodes normally have to be connected a particular way round, for this circuit, it doesn't matter. Using a breadboard isn't recommended because several of the components won't easily plug in, so we suggest mounting the components on a wooden board. It's easy to figure out how to construct your crystal radio this way, or you could build it as in the photo. You can make it as pretty or as

Figure 2 ⬇
Top to bottom: An AM radio signal; negative periods removed by a diode; radio frequencies removed by a capacitor

IMPEDANCE
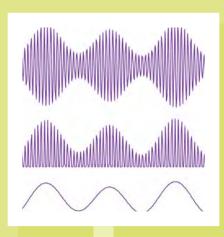
HIGH Q

MEDIUM Q

LOW Q

FREQUENCY

Below ◢
A parallel tuned circuit rejects all frequencies but one, but they're never perfect. Here we see impedance (resistance and reactance) against frequency for three different Q factors

> Put the earpiece in your ear and start tuning the variable capacitor. You should hear one or more stations as you tune

ugly as you want, but do make sure that all the connections are secure. Probably the one thing we do need to say is that you should only connect one end of the coil to the variable capacitor. The connection between the other end of the coil and the capacitor should take the form of a flexible wire lead terminated by a crocodile clip so you can connect it to any tap.
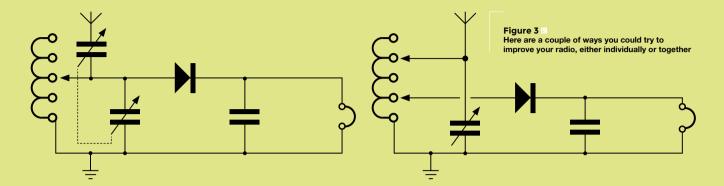
TRYING IT OUT

Once you've built the radio, you need to erect an antenna and arrange for an earth connection, and then connect these to the radio. The length of the antenna depends on the power of the stations you're going to be receiving and how far away they

are. We started by receiving a high-power station at Moorside Edge, which was only 30 km away, using a ten-metre indoor antenna. For more distant or lower-powered stations, you'll need an external antenna. Use plastic-covered stranded wire, make it as long as you can, and attach it at the far end to a tall object like a tree. Make sure it's well-insulated from the support. Use a copper rod, driven into the ground, as your earth – don't use water pipes in the house. Finally, because of the risk of lightning strikes to your antenna, either dismantle the antenna when it's not in use or connect the end to the earth.

Now, attach the flying alligator lead to one of the taps on the coil – if your coil →

and capacitor are like ours, 50 turns would be a good starting point – put the earpiece in your ear and start tuning the variable capacitor. You should hear one or more stations as you tune. If you don't, double-check your wiring, and if that seems OK, either your antenna isn't long enough or there aren't any powerful enough stations nearby. You could check your radio by taking the radio close to a medium wave radio station. If all else fails, use a test meter to check that your variable capacitor doesn't have a short – having first disconnected it from the circuit – especially if it's second hand. If this proves

to be the culprit, you might be able to fix it by carefully bending any plates you can see touching.

HOW IT WORKS

First, a few words on capacitors and inductors. Like resistors, they oppose the flow of an electrical current. This
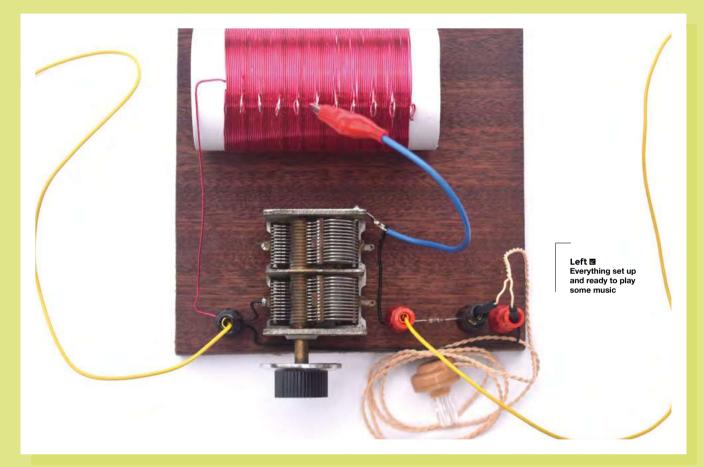
effect is called reactance and, although it's measured in ohms, unlike resistance, it depends not only on the value of the capacitor (in farads) or the inductor (in henries), but also on the frequency. The reactance of a capacitor reduces with frequency, and it increases with an inductor. Where it gets really interesting,

> If you improve the antenna, or listen at night when medium wave radio signals travel further, you'll hear more stations

though, is when a capacitor and an inductor are connected together, to create a 'tuned circuit'. We use a parallel tuned circuit between the antenna and earth. At most frequencies, the reactance is low, so it pretty much shorts the antenna to earth. However, at the tuned circuit's resonant frequency, which alters as the variable capacitor is tuned, the reactance is extremely high. If there's a radio signal on that frequency, despite the power on the antenna being minute, this allows a voltage of a volt or more to progress to the diode.

An AM radio signal is amplitude modulated, which means that the amplitude of the radio signal varies in time with the audio signal – see top part of **Figure 2**. To hear that audio signal, we

need to extract the audio. This is done using the diode, which allows a current to flow in one direction only, i.e. the positive periods – see middle part of **Figure 2**. It's still a modulated radio signal, though, so we need to get rid of the radio frequencies. Because a suitably sized capacitor will pass a radio frequency signal but barely pass an audio signal, a fixed capacitor is used – see bottom part of **Figure 2**. High impedance earpieces often have a degree of capacitance, so the fixed capacitor isn't always needed.

IMPROVING THE CRYSTAL RADIO
The main drawback with the simple crystal radio is that it can be difficult to separate two stations on nearby frequencies, so you'll hear them together. If you improve the antenna, or listen at night when medium wave radio signals travel further, you'll hear more stations, but it'll be even

harder to separate them. Improvements to the basic circuit attempt to improve selectivity without unduly reducing sensitivity. Here are a couple of methods you could try, either individually or together.

The basic circuit uses a parallel tuned circuit, but you can make the radio more selective by adding a series tuned circuit as shown in the schematic (**Figure 3)**. Unlike parallel tuned circuits which are described as rejector circuits, series tuned circuits are acceptor circuits, so putting one in the antenna lead is exactly what we need. Its resonant frequency is the same as a parallel tuned circuit with the same inductance and capacitance, so if you have a dual gang variable capacitor, you can use the second gang for the series tuned circuit, and the two tuned circuits will tune in step with each other.

The other option is to connect the diode to a lower tap on the coil than the antenna

and variable capacitor. The voltage will be lower, but it will add less resistance across the parallel tuned circuit. Because resistance across a tuned circuit reduces its 'Q factor', which is a measure of how sharp it is, reducing that resistance improves the Q and makes the radio more selective. □

Medium Wave

The medium wave band from which you'll be receiving signals with your crystal radio is much lower in frequency than VHF, where FM and DAB signals are found. In Europe, it's 531–1,620kHz; in North America, it covers 530–1,700kHz.

INTERVIEW

Anna Ploszajski

Exploring the bit of the Venn diagram where making and materials meet

Anna Ploszajski (pronounced Por-shy-ski) is a cross-channel swimmer, a materials scientist, a writer, and a breaker-down of barriers to scientific understanding. 50% of the HackSpace editorial team listen to her podcast, *Handmade*, from which has arisen a book: *Handmade: A scientist's search for meaning through making.* Naturally, we wanted to talk to her to find out why we humans do what we do when we turn object A into object B. That's a pretty big question, but if anyone can answer it for us, Anna can. →

INTERVIEW

HackSpace **Hi Anna! You've written a book about making. Before we get on to that, though, we'd like to ask you about something you've been working on in your non-writing life – 4D printing. A while ago we saw a box with a hinged lid; the hinges were fabric, and the box was PLA, so you get the benefits of two types of material in one object. I guess what you're doing is rather more advanced than that?**

Anna Ploszajski You say that, but I've been doing quite a lot of experiments in 3D printing onto fabric to try and make a 4D thing, because PLA has a kind of shape memory. I was wanting to do the experiment that I was doing (which actually I described at the end of the book). I'm trying to draw a conclusion about how my adventures in craft had also impacted my scientific research life. And the example that I use is this experiment that I did 3D printing onto fabrics.

What I was doing began with sort of pre-stressing, just normal textiles. I think there was a cotton, linen, pre-stressing it, just stretching them out with my hands, and then attaching them onto the print bed. And so, you already put in a kind of internal strain into the fabric, then 3D-print a very simple design that was either a circle, or just simple lines. And then obviously, when you print onto it, then the PLA plastic is bonded onto the textile. My idea was that if you then heated that material up, then it would soften, and that tension that you'd put into the fabric would be released. So that was my idea.

My project was all to do with exploring this idea of 4D printing. So printing, using 3D printing, to make objects that move in some way after you've printed them. The thing about it is, it's adjacent to this topic of smart materials. There's a family of materials that have some kind of smart property, usually it's colour-changing or shape-changing in response to an external stimulus. So, that could be

temperature change, or light levels or moisture levels.

And those smart materials are not actually that smart, it turns out, because what they do is really simple. Let's take the example of a really simple shape change: wood is a really good example. It expands when it gets wet. And it contracts when it dries out. By our definition of a smart material, that is a smart material because it changes shape when there's a change in environment. And that's a very simple movement. And these smart materials tend to just have this kind of flip-flopping between two simple states – either, you know, an expanded state or a contracted state in this example. That's not actually that useful, unless you can do a clever design to use that movement to form a clever kind of motion.

A really good example in nature is the pine cone; the spines of a pine cone have this really ingenious bi-layer structure,

> " I wanted to write a popular science book that didn't cater to the normal popular science audience

where one side of them has a very hygroscopic wood – it expands a lot when it gets wet. And the other side doesn't expand a lot when it gets wet. So, when the pine cone gets wet, it's that bi-layer structure that causes that movement. The wood itself is just expanding. But the contrast between the two is what causes that motion. So I was trying to get inspired by that and combine, using clever design, a quite simple, smart material with some design that would combine it with a non-smart material that would cause some kind of motion.

It's all to do with stored tension, and triggering that tension release. And to be honest with you, I didn't get very far with

it. I understand the material side; that was fine. And I could do all my experiments in the lab, and I could characterise the materials fine, but I just don't have a designer's brain.

And that is what the book is about in a way: trying to access or tap into these other skills that designers and makers and craftspeople have which I don't.

HS **How much have you learned over the course of writing the book? You must have had to speak to all sorts of people to research it.**

AP I think that meeting all those craftspeople, and getting a view into their world, really gave me an appreciation for exactly how much work and time and skill and practice goes into really honing these skills. Wood is a really good example: when I did the wood carving workshop with Barn the Spoon, it took hours trying to make a spoon, but when I did it, mine didn't look anything like his spoons.

The skills themselves are often not that complicated or difficult to do. It's the constant practice in refinement and design, which are the skills that I didn't necessarily have.

HS **What led you to write the book?**

AP A few things. Firstly, I wanted to write a popular science book that didn't cater to the normal popular science audience, by which I mean people who are already relatively interested in science, the types of people who would browse the popular science sections in a bookshop and pick things up about space, or the gut, or whatever. I feel like that audience is already very well catered for.

What I wanted to do was try and write a popular book that would be read by someone who would never normally read a science book – that's the whole of the rest of the population. So you'll notice in the book that there are a lot of scientific →

analyses and explanations, but they're all quite short. And my hope was that, if someone's coming at this with not very much prior knowledge of science, they get to a description of the quantum mechanics behind why glass is transparent. But on the next page, we're back to the story. And it's really those stories that were the most important thing to me.

And so, in each of the ten chapters on different materials, the story isn't the story of the material – it's the story of something else. So in *Plastics*, it's the story of my Polish grandad and, you know, his life story throughout the 20th century, which intertwines with the story of the rise and fall of plastics.

I wanted to draw all these other audiences in by storytelling, and then hopefully, sneak the science in when they weren't looking.

The story of the book itself is to do with feeling very inadequate, I suppose. I had this realisation, having walked into the Institute of Making for the first time, that I was supposedly this expert in materials, having studied the science of it, having studied all on paper, but actually, there were all of these different people that had so much more in-depth knowledge than me. The craftspeople and the makers and the artists and the historians and the designers and the architects… And so it was them that I really wanted to spend time with and learn from.

That was four years ago. That was when I started my podcast, which is also called *Handmade*. And that was where I started interviewing makers and craftspeople. And the book just grew from that. Quite a few of the people that I interviewed on the podcast have ended up being featured in the book as the very, very, very kind craftspeople that took me under their wing and showed me the ropes of what they do.

To take blacksmithing as one example – I thought I was an expert in materials, but I had never felt metal softening under

my fingers. Yes, I knew the theory, I could draw you the iron-carbon phase diagram, I could talk about the phases and melting, and all of the ways that carbon and iron interact at the atomic level inside steel. But I've never done it. And I didn't know how hard you had to hit it to make it change shape. Agnes, the blacksmith who taught me, is just so, so brilliant. I'm such a huge fangirl of her. And it was very humbling, actually, to spend time with people like that.

HS **Getting to touch and feel the materials rather than study them, was there any one in particular that you gained an appreciation of?**

AP My favourite chapter in the book is *Sugar*, because it was the most fun story to write. And it's the story of my English Channel swim. [Yes, you read

> I think that's what craft gives us – it's a sense of purpose almost, and a sense of achievement at the end

that right – Anna has swum the English Channel.] One of the reasons was, I think, it already is one of the strongest chapters for storytelling. Because it is this kind of archetypal physical journey from A to B, but also a journey of discovery about yourself. And intertwined in that story is the story of sugar, and all its different forms, and how it affects the body and the mind.

In terms of the crafts, it was really wool that caught my imagination, and I've stuck with it. The story of wool is the story of my camper-van trip around Scotland and the north of England. I acquired wool from all these different places that I went to on my trip, and then knitted a patchwork blanket with all the wool I got

from the different places. And through doing that, I taught myself how to knit and I met all of these kinds of amazing knitters and wool-craft people throughout Scotland and the north of England, and chatted to them and got an insight into this amazing world of women who knit – and they were all women – and what it means to them, and how it connects them. And it's very meditative, I find, and that's the craft that I've taken through since finishing the book a year ago. That's the craft that I've continued with.

I don't know what it is about it. It just feels so nice to create something, you know, especially in the last year when we were all sitting at home watching Netflix and trawling through the movies and TV shows on there. Although that felt like perhaps a bit of a waste of time, actually, if I was knitting while watching TV, it wasn't all a waste of time; I had something to show for it at the end. And I think that's what craft gives us – it's a sense of purpose almost, and a sense of achievement at the end.

You know, to have that sense of achievement of 'I've made this' and now I can wear it, or now I can use it. I haven't had that in science before. I only got that when I started entering this world of craft.

HS **It sounds like you see a disconnect between science and making. Is that fair to say?**

AP I've thought a lot about this: this kind of compartmentalising of making and science, or art and science as I talk about in the book (and I know that art and making are absolutely not the same thing). And I think there are a lot of reasons why the arts and sciences have been sort of severed from each other. In formal education, we separate them. At school, we have to often choose between those types of subjects. I ended up going down the science route, but I did A-level music. I love writing and music and history, and I was always crap at art, but I enjoy it. I think it's really unhelpful that →

Image Charlie Murphy

we do that, because it means that we brand people as 'you're a scientist', or 'you're more of an artist'. And actually, I think the majority of people are probably somewhere in the middle. Actually, they have interest in both.

It's a real shame that we often get siphoned off into these different camps, and often don't get the chance to rediscover the other one. As someone who was siphoned off into the scientific track, it was really liberating to be able to discover the craft and artistic world. It was, like I say, very humbling. It was also really nice to be a complete beginner again at something, to be able to ask the silly questions from a place of curiosity, with no pressure, no educational pressure. I wasn't trying to achieve anything apart from trying to make a spoon, or forge a knife, or throw a pot, or whatever it was.

Materials is a really interesting subject because it can sit at this intersection between the artistic world and the scientific world. Materials, perhaps uniquely in the sciences, is a really lovely way to explore the more artistic side. And what I've discovered through the book and through the podcast, is that we all

understand these materials, maybe in slightly different ways. But quite often, it's just that we use different language to talk about them. I remember interviewing a silversmith on my podcast called John Cussell, who described cold-working silver metal to me as making the atoms angry. So, when you cold-work silver, it becomes more and more stiff. I would describe that as putting dislocations into the material and putting internal stresses and strains to make them more brittle. We're both talking about the same thing

in different ways. And I think that, really, the wonderful thing that I love about materials is that it can be this common substance, literally, through which all sorts of different people can talk to each other.

HS This issue, we're talking about citizen science, which has taken huge steps forward recently in broadening access to scientific research, but very often it's locked away inside university buildings and it's a real shame. What do you think can be done about that?

AP That's my life's mission, to try and break science out of universities through doing things like writing the book and the podcast, and the talks that I give. I really want to invite people in and show them that science – it's a huge cliché but science really is everywhere. It's never been more important than in the last 18 months to understand science, virology, how contagions spread – that's all science. And the science communication that's going on around that has been mixed. Some of it's been really good, but

Above
It's one thing to understand the molecular changes that occur when you fire clay; it's another thing entirely to be able to make a pot

some of it's been really damaging. That's what's important to me is to break science out of these institutions, because a lot of people are turned off science at a very early age. And unlike a lot of other areas, it's impossible to turn back. If you go down a non-scientific route, through school, and then maybe through university or through a job, it's impossible

to go back on that and pick it up again later. I feel like subjects like history and literature are much more accessible to everybody. Whereas science is considered to be more for a select few, you know, a chosen few who are allowed to do it. And that's really not fair.

HS Are craftspeople scientists? There must be a lot of crossover in terms of learning, experimentation, and so on.
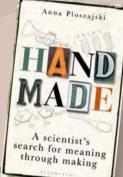
AP I think you'd have to ask them, but whatever it is they do is experimentation, right? And they do experiments all the time – what temperature do I need to make my steel to make it do X? Or, what composition do I need my clay to be to make it do Y? How do I do the settings on my furnace to make sure that my pots don't explode? And that is exactly the sort of stuff that we would do in the lab, you

know: methodical experimentation. So in that way, definitely. I can't see that there's any difference at all between that. And in terms of the way that craftspeople and scientists think, that's much more difficult to answer.

Most science has arisen from craftspeople and early experimenters. The subject of material science arose out of the subject of metallurgy, which arose out of blacksmiths like Agnes. If you go back far enough, it's all the same thing. □
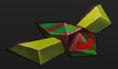
Handmade: A scientist's search for meaning through making **by Anna Ploszajski (Bloomsbury Sigma) is out now.**
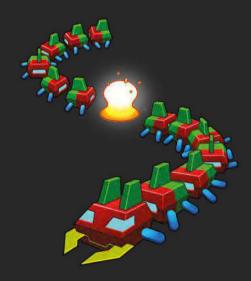

Anna Ploszajski

HAND MADE

A scientist's search for meaning through making

Right
Wool was hugely important for England's development into a major mediaeval power. It's also good for keeping your feet warm

NAILS

IMPROVISER'S TOOLBOX

Every shed, garage, and workshop, everywhere, has a big box of assorted nails sitting idle. Andrew Gregory finds a few ways to put them to work
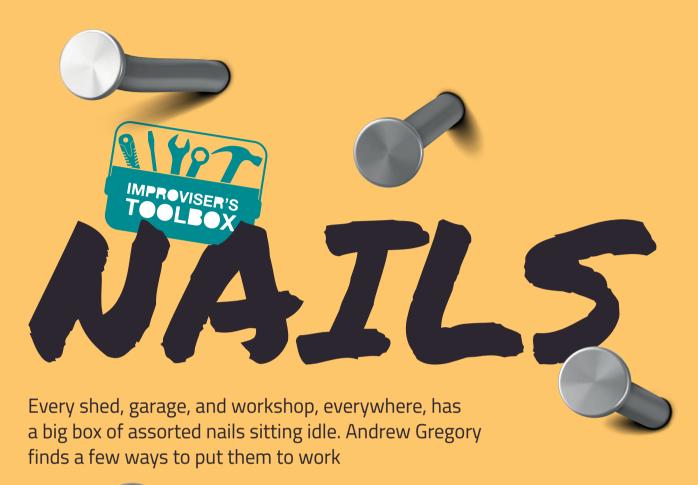
Andrew Gregory

🐦 @AndrewGregory83

Andrew dabbles in Raspberry Pi, music and woodwork. So far his most impressive build is a pile of sawdust and a load of very sharp chisels

f all the connectors that we use to join one bit of wood to another bit of wood, the simplest is the nail. Like its close cousin the dowel, nails have been used for thousands of years. Although we're used to seeing steel or iron nails, nails have been around for far longer than we've been using iron: early nails made of bronze have been found in Egypt dating back to around 3400 BC. That makes them significantly older than their functional equivalent, screws, which had to wait for iron, files, and other tools to be invented before they could be made.

Nails available today are either wire nails, made from extruded wire which is cut into lengths, made pointy at one end and flattish at the other end, and cut nails, which (as the name suggests) are cut out of a sheet of metal. The method for making cut nails is older than that for making round nails, and it throws up an example of where improvements in the manufacturing process (round nails are vastly cheaper to make) result in an inferior end product. For woodwork, at least, cut nails give a much stronger join, as they resist rotational force better, and are harder to pull out once they're hammered in. For this reason they're sometimes used in furniture making, but for most people, the slight benefits will never outweigh the increased cost.

Apart from shape and size, the other variable with nails is the material they're made from. Steel is most often used, either uncoated (sometimes referred to as 'bright') or coated, and you'll also find stainless steel, copper, iron, bronze, and other materials – including brass nails, used in place of ferric metals where you wouldn't want to accidentally make sparks, such as on the deck of an 18th-century warship. That's all by the by though – nails are so standard nowadays that they're sold by weight, meaning you probably have some rolling round in a drawer or cupboard somewhere. Here are a few ideas of what to do with them.

"EARLY NAILS MADE OF BRONZE HAVE BEEN FOUND IN EGYPT DATING BACK TO AROUND 3400 BC"

NAIL KNIVES

Mario Cesari teaches artisan metalworking in a small town called Pennabilli, up in the hills that run down the middle of Italy. This hack with nails is on the smaller end of his skills, but it's impressive to us. With such small pieces as these, you don't need an anvil on which to beat the metal: a larger hammer will do. Iron nails can be beaten into shape cold, but if you have mild steel nails, you've got to heat the metal until it's red, then let it cool down slowly in the air to anneal it.

The blade needs to be thicker at the back, to give it strength, than it will be at the cutting edge, and because of this, it will naturally take on a curved edge as you're hammering it flat. To counteract this, Mario bends the nail slightly before he starts flattening the blade, which straightens out when he starts to work it. You end up with a tiny knife that can take an edge and is ideal for scoring lines in card, opening letters, marking where joints in woodwork need to be cut, or anything else that needs a small, sharp blade. →

Project Maker
MARIO CESARI

Project Link
hsmag.cc/NailKnife

> "YOU END UP WITH A TINY KNIFE THAT CAN TAKE AN EDGE AND IS IDEAL FOR SCORING LINES IN CARD, OPENING LETTERS..."

Left ⬔
As nails are cheap, you can experiment with different blade shapes

STRING ART

Project Maker
HELEN

Project Link
hsmag.cc/StringArt

—

For this project, maker Helen used a piece of plywood 60 × 40 cm, 200 g of nails, and around 200 m of string. She started with the outline of the world map printed out on paper, carefully sellotaped to the plywood base (you could always use a map, or indeed any other design you want to replicate). She then went around the outline of the continents and islands, hammering in a nail every 1 cm to form the outside edges of the shapes. After that, the amount of infill you use is up to you.

Helen stopped when she felt that the artwork was complete, but there are loads of ways this build could have been different: more string inside land-masses would give a stronger outline to the Mediterranean and the Gulf of Mexico; you could use different coloured string for different continents; more or fewer nails to get more detail, or individual nails to mark out points of interest within a land-mass. It's an infinitely customisable idea, and that's why we like it.

Right ⬦
For a different approach, you could paint the plywood dark, and use brightly coloured string

MAGNETIC BOTTLE OPENER

It's important to stay hydrated, especially now that we're making the planet too hot to sustain human life. Rather than open beer bottles using a polished worktop, teeth, or keys, Mike Warren treated himself to a bespoke cap remover incorporating a couple of extra usability enhancements.

The business end of the bottle opener is a two-inch nail – crucially, this has to have a flat head, as bullet-headed or hidden head nails won't give you any purchase under the bottle cap. This is bent and inserted into a shaped piece of scrap wood. That would be enough for a minimum viable bottle opener, but Mike has gone further, shaping the wood with a rotary

Project Maker
Mike Warren

Project Link
hsmag.cc/BottleOpener

tool and 80-grit sandpaper and staining it with tea, vinegar, and steel wool. Finally, Mike's also used epoxy resin to embed a couple of neodymium magnets into the wooden handle: one to keep hold of the bottle top once it's been taken off, and one to stick it somewhere safely out of the way when he's not using it.

NAIL WALL

We've seen versions of this project made using pins, but this one uses nails, which, thanks to their larger size, means the process is relatively quick. The materials involved are a piece of plexiglass, a piece of wood/plywood, some three-inch nails, four six-inch bolts, to space the plywood out from the plexiglass, and enough nuts to keep the bolts in place (this build uses 16 nuts).

Rather than laboriously drill each hole by hand, Rimamonsta used a laser cutter to cut out a pattern she'd drawn in Inkscape. It was a limitation of the laser cutter that led to nails being used in this build rather than pins – for whatever reason, the laser cutter wouldn't cut out holes small enough for pins, so the holes had to be enlarged and nails introduced

rather than pins. If you're inspired to follow any of these builds, please be aware that when they're fresh out of the box, nails sometimes have a light coating of grease that you'll need to wipe off before you put them anywhere near a decorative surface.

Project Maker
Rimamonsta

Project Link
hsmag.cc/NailWall

Above
Mike keeps his bottle opener stuck to his fridge door

Left
This build was inspired by a visit to the Exploratorium's gift shop in San Francisco

IN THE WORKSHOP:
Making T-shirts
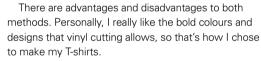
By **Ben Everard**   Vinyl is making a comeback

If you want to make T-shirts at home or in a small workshop, there are two main ways you can go about it: sublimation printing and vinyl cutting. Sublimation printing works by using a special printer to print onto special paper, and you can then use a heat press to transfer this image to a T-shirt. Vinyl cutting uses a machine called a vinyl cutter (which is basically a CNC knife) to cut out a design from thin vinyl. You can then use a heat press to permanently adhere this vinyl to a T-shirt.

There are advantages and disadvantages to both methods. Personally, I really like the bold colours and designs that vinyl cutting allows, so that's how I chose to make my T-shirts.

I got some plain T-shirts and a selection pack of different colours of heat-transfer vinyl (also called T-shirt vinyl). The exact process you'll need to follow will depend on your particular hardware, but here's how I did it.

The first stage is the design. As we're working with (basically) a CNC knife, it should come as no surprise that you need vector artwork. This means art defined by a path detailing the shape of the artwork, rather than detailing pixel-by-pixel what the image should contain (if you have the latter – known as raster images – you may want to investigate sublimation printing).

I obtained my vector images in a couple of ways. Some I found with an online search for vector images (you can search for images which are licensed for reuse here: **hsmag.cc/vector**). Some I found online as raster images, but with simple outlines (I searched for Silhouette), and then traced them. The text, I could just write.

I have access to a Silhouette Cameo 2 at my local hackspace. It's not the most recent model, but it works reliably.

My workflow is slightly clunky: Inkscape (for image processing) which outputs DXF files for Silhouette Studio. This then creates the .studio files which I can load to a USB stick for transfer to the cutter.

The first step in Inkscape was to convert everything I wanted to cut into a DXF file. With text, I typed it out, selected the font, size, style, etc. Once done, I used Path > Object To Path to create an outline that could be cut, then Save As and selected DXF. I wanted each part to be cut in a separate DXF file as I cut each colour separately, and might want to cut each part of the same colour separately so I could position them independently on the T-shirt. This means that in some cases I split the words up into different DXF files.

With vector images, the process is usually just opening it in Inkscape and then saving as DXF.

For any silhouette images, I imported them into Inkscape then used the Path > Trace Bitmap feature to trace a line around the edge. This traced line is what will be cut. Again, once this is done, I save as DXF.

Once I'd got all the DXF files for a design, I opened up Silhouette Studio. I needed a separate project for each colour, but multiple bits of the design can be in one project if they're in the same colour. They can be cut apart and placed separately, so they don't need to be in the right position – you can place them closer together to save space.

The easiest way to waste time and vinyl is to forget to flip the image horizontally. As you cut one side of the vinyl, then flip it before placing it on the T-shirt, you need to flip it before you cut it. Right-click each item and select Flip Horizontally.

If you haven't knackered your USB setup, you can connect your vinyl cutter to your computer and then click Send.

However you get your design to the vinyl cutter, you need to enter the correct cut settings. There are several preset options and vinyl is one of them. Before doing your first cut, it's worth doing a small experiment first to make sure that it cuts properly. Properly means that it cuts through the vinyl but not the backing. You should be able to peel away the vinyl you don't want, leaving behind the vinyl you do, and the two pieces should separate cleanly.

The second easiest way to waste time and vinyl is to cut the wrong side. T-shirt vinyl has a shiny side and a matt side. It's the matt side that should be face-up in

the cutter. Depending on your cutter, you may also need a cutting mat.

With everything set up, just start the cut and set it to work. After a few minutes, you should have a nicely cut piece of vinyl.

With a pair of scissors or a craft knife, you can cut around the area your designs are in. You don't need to be too accurate, just make sure you have the whole of your design in the area you cut out.

Now, you need to remove the excess vinyl. Depending on your design, this might be easy or difficult. It helps to have a 'weeding tool', but you can probably get by with some bent wire or a pair of tweezers if you don't have one. There's no great secret to this: just pick away at the bits you want to remove and pull them out.

Finally, you just need to press and heat the design on. You can use an iron for this, but I used a heat press and some PTFE sheets to protect the T-shirt. Once it's up to temperature, one short press got the T-shirt flat and fully dry. You place the piece of the design matt-side down on your T-shirt. It's at this point that you may discover that you forgot to flip the design horizontally before you made the cut. I won't admit how many times I did this, but it was more than once. A final press of 15 seconds (or whatever your vinyl manufacturer recommends) fixed the vinyl onto the T-shirt. Once everything's cooled down, you can pull the clear backing off the vinyl and your T-shirt is ready to wear. ◻

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE
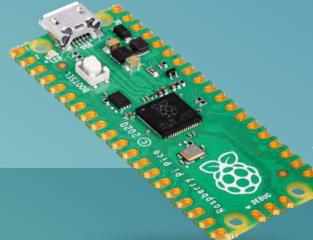
HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

Make music with Pico

Send MIDI signals to your computer or other MIDI systems

Ben Everard

🐦 @ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Below ◈
You can use any analogue input, but we used a potentiometer for simplicity

MIDI – Musical Instrument Digital Interface – is a protocol that lets, well, musical instruments interface digitally. Basically, it's a way of sending m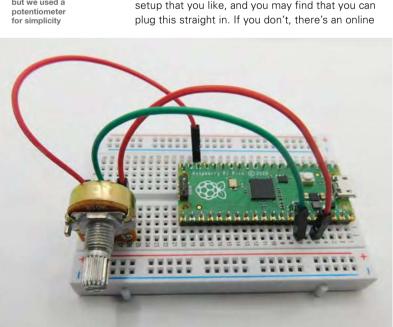essages back and forth between keyboards, synthesizers, computers, and other noise-making machinery. It's been around since the early 1980s and is supported by a huge range of hardware. While it is standard, there are a few different physical connections. We're going to be using USB here, but there is also a specific connector just for MIDI that some older hardware may expect.

There are quite a few different types of messages you can send, but the ones we'll look at are note messages (which start and stop notes) and control messages (which you can use to alter some part of the setup).

We're going to be making a MIDI controller. That means that it doesn't create sound, but it can be used to manipulate a synthesizer. If you already tinker with digital music, you probably have a MIDI setup that you like, and you may find that you can plug this straight in. If you don't, there's an online

synth at **nicroto.github.io/viktor**. Click 'Load Synth' on that page to get a playable synth in your web browser. Mash the keys or click around with the mouse to start making noise.

We're going to use CircuitPython as this has support for USB MIDI built in. If you haven't got it already, you can grab the latest version for Raspberry Pi Pico from **circuitpython.org** and flash it to your Pico (you'll also find instructions on how to flash the firmware if you've not done this before).

You'll also need some libraries. You can download the bundle from **circuitpython.org**, extract **adafruit_midi** and **simpleio**, and copy them to **/lib** on the CircuitPython USB device.

Finally, you'll need a CircuitPython editor. We use Mu. If you've not used it before, there's a guide to getting started at **hsmag.cc/InstallMuEditor**. Now we're up and running, let's play a little tune. Load the following code onto Pico (you can download it from **hsmag.cc/Randomtune**):

```
import time
import board
import busio
import usb_midi
import adafruit_midi
from adafruit_midi.note_on import NoteOn
from adafruit_midi.note_off import NoteOff
import random

USB_MIDI_channel = 1  # pick your USB MIDI out
channel here, 1-16

usb_midi = adafruit_midi.MIDI(
    midi_out=usb_midi.ports[1], out_channel=USB_
MIDI_channel - 1
)

counter = 0
current_note=10

while True:
    # read all the knob values

    counter = counter+1
```

```
    if (counter == 100) :
        usb_midi.send(NoteOff(current_note, 10))
        current_note = random.randint(0,100)
        usb_midi.send(NoteOn(current_note, 10))
        counter = 0

    time.sleep(0.01)
```

With the code running, refresh the synth web page and press Settings, and you should see your MIDI device listed. If it's not, unplug and replug in your Pico and try again. If everything goes well, you should hear a random tune. Well, tune might be a bit much, but a series of notes at least.

That's the music playing, but what if, rather than play notes, we want to manipulate the sound that the synthesizer makes? We can do this using control changes.

In the below code, we're using the analogue input A2 on Pico, mapping it to the range 0–127, and sending it to ControlChange 1 (which is the mod wheel). You'll need a way of getting some input into A2, and we used a potentiometer. There should be three pins. Connect one end to 3 V, one end to ground, and the middle pin to A2. Load the code from **hsmag.cc/ModWheel** and you should hear the sound change as you twiddle the knob. The important parts are:

```
knob = AnalogIn(board.A2)
cc_number = 1
cc_range = (0, 127)
cc_value = (0,0)
last_cc_value = (0,0)

def range_index(ctl, ctl_max, old_idx, offset):
    if (ctl + offset > 65535) or (ctl + offset <
0):
        offset = 0
    idx = int(map_range((ctl + offset) & 0xFF00,
1200, 65500, 0, ctl_max))
    if idx != old_idx:  # if index changed, adjust
hysteresis offset
        # offset is 25% of the control slice
(65536/ctl_max)
        offset = int(
            0.25 * sign(idx - old_idx) * (65535 /
ctl_max)
        )  # edit 0.25 to adjust slices
    return idx, offset

def sign(x):  # determine the sign of x
    if x >= 0:
        return 1
```

```
    else:
        return -1

counter = 0

while True:
    cc_value = range_index(
        knob.value,
        (cc_range[1] - cc_range[0] + 1),
        cc_value[0],
        cc_value[1],
    )
    if cc_value != last_cc_value:
        usb_midi.send(ControlChange(cc_number, cc_
value[0] + cc_range[0]))
        print("sending change")
        print(cc_value)
        print(last_cc_value)
        print(knob.value)
        last_cc_value = cc_value

    time.sleep(0.01)
```
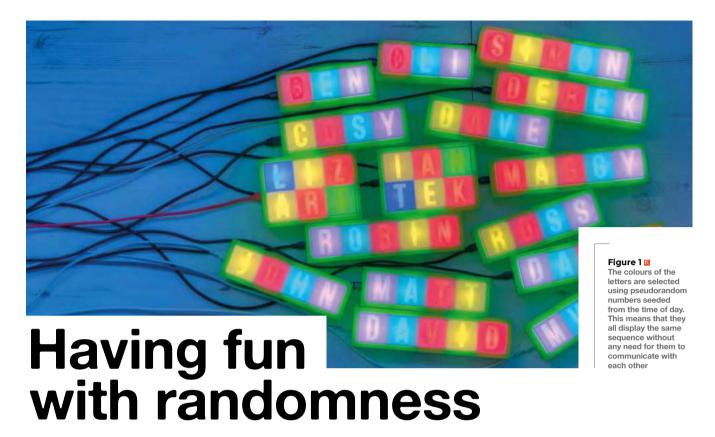
This really has been a crash course in MIDI with Pico, but even with these simple techniques, you can create some fantastic music controllers. You can give yourself physical knobs and sliders to avoid finicky on-screen controls, or you could even hook up environmental sensors to create sounds that change depending on the air quality, temperature, or other parameters. ◻

STOLEN CODE

This is a simplification of one of John Park's projects where he uses an Adafruit Grand Central M4 Express to hook up 16 knobs and feed them all in through MIDI. If you want to take this to the next level, take a look at his project at **hsmag.cc/USB_MIDI_CircuitPython.**

Figure 1 🔗
The colours of the letters are selected using pseudorandom numbers seeded from the time of day. This means that they all display the same sequence without any need for them to communicate with each other

Having fun with randomness

Making synchronised coloured lights using the magic of pseudorandom numbers

Rob Miles

🐦 @robmiles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at **robmiles.com**.

In this article we are going to look at randomness, and discover how to create it using software. Then we'll investigate different flavours of random numbers, and finish by using managed randomness to create a light that displays colours which can be synchronised worldwide.

Figure 1 shows the light-boxes that we are going to make. Each one contains a strip of NeoPixels and an ESP8266 device. They are running the Connected Little Boxes software, which you can find here: **connectedlittleboxes.com**. You can also use the software to control chains of LEDs, as well as names with lights in. Let's discover how random numbers make this work, and then look at how the lights are constructed.

MAKING RANDOM NUMBERS IN PROGRAMS

Most programming languages provide a way for programs to obtain random numbers. Let's start by looking at how a program running on an Arduino device can get hold of random values and use them to create a simple dice application.
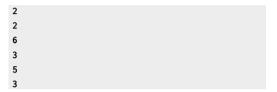
```
void setup() {
  Serial.begin(115200);
  int randomNumber;
  for(int i=0;i<6;i++){
    randomNumber = random(1,7);
    Serial.println(randomNumber);
  }
}

void loop() {
  delay(1);
}
```

The C++ program above runs on Arduino devices. It uses the `random` function from the Arduino libraries in a loop to print out six random numbers in the range 1–6. Note that the second parameter to the `random` function (the value 7 in the code above) is an exclusive upper limit, meaning that the value 7 will never be generated as a random result. This program

prints six throw values and then stops. If you load this program into a device, and look at the terminal output it prints, you'll see the values:

```
2
2
6
3
5
3
```

The numbers above were produced by running the program on an Arduino Uno. If you run this program on your own Uno, there is a good chance that you will see the same sequence of values. If you run this program on a different type of Arduino, perhaps an ESP8266 or ESP32, you will see a different sequence. It turns out that some devices produce the same sequence each time they are reset, whereas others print a different one. The question we now must consider is: 'What is randomness, and why is it sometimes the same?'

RANDOM GENERATION

Hardware manufacturers work very hard to make sure that a program does the same thing every time it runs. A computer that behaves in a random way is usually described as 'broken'. The random number generator used by the Arduino is called 'pseudorandom'. It produces results that appear random, but are actually a sequence of calculated values. The next random value in the sequence is produced by using the previous value.

> **"** You can make **pseudorandom number** sequences that don't repeat for a very long time **"**
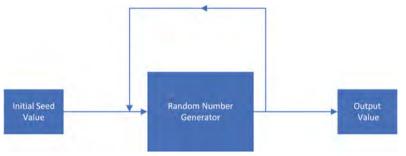
Figure 2 shows how this works. An initial 'seed' value is fed into the generator which works on it to produce the first random number. Then, each successive value is produced from the previous value. There are lots of different ways to generate each number. We'll look at one called a 'linear congruential generator'. This is a very impressive name for 'multiplying the previous random number by a big constant, adding another big constant, and then wrapping the value round inside a third big constant'. By careful choice of your 'big constants', you can make pseudorandom number sequences that don't repeat for a very long time. The code to implement it looks like this:

```
int rand_value=1234;
int rand_mult=8121;
int rand_add=28411;
int rand_modulus=134456;

int rand()
{
  rand_value = (rand_mult * rand_value + rand_add) % rand_modulus;
  return rand_value;
}
```

The function **rand** returns a new random number each time it is called. The variable **rand_value** holds the current random number. It is initialised to the seed value 1234 when the program starts, which reflects something of a lack of imagination on the part of the author. The **rand_mult** and **rand_add** values are applied to the current **rand_value** to make the next one when the **rand** function is called.

The value in **rand_modulus** determines the range of the random numbers produced by the generator. →
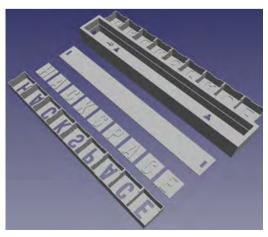
Figure 2 ◈
The random number generator generates a new value by doing something to the previous value

Figure 3 ◈
There are two designs for a light-box. You can 3D-print transparent letter tiles which fit into an opaque top panel and/or a single piece top panel that has thinner areas for the letter's shapes to show through. You can also design boxes that contain multiple rows of text

Figure 4 ↗
The box contains mounting pillars for a clip that holds the Wemos D1 mini ESP8266 device in place. The micro USB connection on the Wemos is exposed through a hole in the box so you can connect the box to power

The modulus operator means 'give me the remainder when you divide by this number'. In other words, the value of 250% 100 (250 modulus 100) would be 50, because 100 goes into 200 twice and leaves a remainder of 50. We need to apply this modulus operation to stop our random number value from getting bigger and bigger each time **rand** is called.

If we want a random number in a particular range, we can use the modulus (%) operator on the value returned by the **rand** function:

```
int throw_value = (rand()%6) + 1;
```

The statement above would set the variable **throw_value** to a random value between 1 and 6. The

PSEUDORANDOM GAMES

We can use a pseudorandom sequence to generate assets and behaviours in a game. Different values might mean that a location on the map is sea, land, or mountain. Or, we could use the values to control properties of characters in the game. Pseudorandom sequences are a great way to get complex, repeatable environments without having to store any of the data that describes them. Game creators have been using this technique ever since the hugely popular Elite game appeared on the BBC Micro home computer back in 1984. Elite uses sequences of pseudorandom values to generate the names and attributes of all the worlds the player can visit. You can investigate how this works at **bbcelite.com**. This technique was brought bang up to date by the makers of the No Man's Sky video game released in 2016, which uses pseudorandom techniques to make 18 quintillion (18 followed by 18 zeros) different planets. All the players of these games have exactly the same experience because the random number generators used are seeded from the same values.

modulus operator gives the remainder after dividing the random value by 6 (which will be in the range 0 to 5). The statement then adds 1 to it, producing a result in the range 1 to 6.

PLANTING A SEED

Now that you know how the random number generator works, you can see how we can get the same random number sequence each time a program runs. Successive calls to the **rand** function will generate the same sequence of numbers for a given initial value of **rand_value**. In the case of the code above, the initial value is 1234. We say that 1234 is the seed for our random number generator. The Arduino Uno uses the same seed value each time it is reset, so you see the same sequence of random numbers when the dice program runs. The ESP32 device uses radio noise from the WiFi and Bluetooth receivers to create a random seed, which is then used by the random number generator on those devices. This means that each time an ESP32 is reset, it will generate a different sequence of values. If you want your device to use a particular seed value, you can use the function **randomSeed** to set that value:

```
randomSeed(1);
```

This would set the initial value of the seed to 1, producing a particular sequence of random numbers. If you are using an Arduino Uno and you want to have different random behaviour each time your program runs, you must find a way of obtaining a random value to feed into the **randomSeed** function. One solution is to read a value from an analogue input that has nothing connected to it.

```
randomSeed(analogRead(0));   // seed the time from
analogue input 0
```

This is a 'homebrew' version of the solution used by the ESP32. It uses the analogue input circuit on the Arduino rather than a radio receiver. The idea is that random voltages induced into the circuit by the environment around the device will result in the function **analogRead** returning a different value each time it is called. This might work well for you, but sometimes it seems that the input value gets stuck, reducing the randomness a bit. Another way to get a random seed is to use elapsed time. If your application waits for inputs from the user or another device, a program can use the time since the device was started to generate a seed value:

```
randomSeed(millis());       // seed the time from
the clock
```

RANDOM **SECURITY**

Pseudorandom number sequences play a big part in keeping the internet secure. Sensitive data streams can be protected from prying eyes by using successive values from a pseudorandom number generator to encrypt the contents of a block of data. The data can be decoded if the reader has the same random number generator code and knows the seed value. The random number generator that we are looking at was built for speed, not security. There are 'crypto secure' random number generator routines that you can use if you are really serious about keeping your secrets.

The code here seeds the random number generator using the time in milliseconds since the device was switched on. This is especially useful in adding randomness to games that ask the user to press a button to start the game. Unless your user has ninja reactions that are precise to a thousandth of a second, it is impossible for them to generate the same sequence of random numbers.

RANDOM SEEDING WITH TIME

The software we are going to use in our lights uses the time of day in minutes to seed the random number generator which determines the colours of our lights. This synchronizes the colours for all the lights at any given time.

The Connected Little Boxes software uses the ezTime library (**github.com/ropg/ezTime**) to obtain the date and time from a network connection. It then sets an internal reading structure (`clockActiveReading`) with values received from the clock.

```
unsigned long seedValue = (clockActiveReading->hour
* 60) + clockActiveReading->minute;
localSrand(seedValue);
```

The statement above shows how the `localSrand` function is used in the Connected Little Boxes code to set the seed value for a local random number generator using the hour and the minute values from the clock. We don't have to work with this code, however; we just have to tell the box what to do:

```
{"pro-cess":"pixels","command":"twinkle","option":
"timed","sensor":"clock","trigger":"minute"}
```

This Connected Little Boxes command tells the pixels to twinkle with random colours which are seeded from the clock. The command is triggered by the clock sensor and updates the colours every minute. You send the command to the box and then you can watch the light display change every minute.

BUILDING BOXES

Now that we have our software working, we can look at the hardware for our 'Lights in Names' devices. The box designs are 3D-printable and are produced by a Python program running inside the FreeCAD design tool. You tell the program the text for the light, and it then creates a box design (**Figure 3**). You can find full instructions on how to install the Python script and use it to create designs here: **hsmag.cc/LightsInNames**.

The LED strip is stuck onto a backing plate which then slots into the case, as shown in **Figure 4**. The front panel is then pushed into the case and held in place with some spots of hot glue. Wire-wrap wire is used to connect three pins from the Wemos device to the strip of NeoPixels (**Figure 5**).

FUN WITH LIGHTS

The light-boxes can be used to create synchronised lights so that even though you are a long way from a loved one, you can know that they are seeing the same coloured display as you. You can change the colours on the hour or daily if you don't want too many changes. The author hopes you have fun with them. □
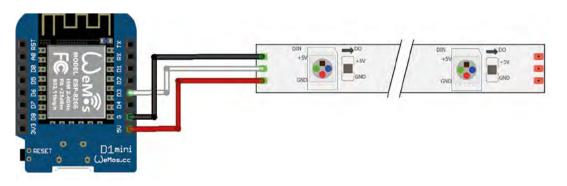

Figure 5 ◆
The Wemos device is powered via the USB connection. This will work well for the small number of lights that we need for a name. The connections are made with wire-wrap wire, which is great for soldering onto the tiny contacts on the LED strip

MAKER

K.G. Orphanides

K.G. is a writer, maker of odd games, and software preservation enthusiast. Their household can now hold very retro Street Fighter II tournaments, and that's beautiful

@KGOrphanides

Part 05

Build an arcade machine: RetroPie and stream from Steam

Use RetroPie as your arcade operating system and add extra emulators with support for Steam Link. Stream games from a powerful PC to Rasbperry Pi

Last month, we used Recalbox for our **main arcade cabinet operating system, but it's not your only choice.** In this final instalment of the 'Build an arcade machine' series, we'll use the RetroPie distribution, currently at version 4.7, to provide extra features such as Steam Link support, as well as taking a longer look at where to buy arcade games and how to get them onto your system. This tutorial assumes that you already have a fully assembled and wired arcade cabinet.

▼ More button assignments are available on RetroPie than you have arcade controls. You can skip the ones that don't match up

01   Install and prepare RetroPie
Fire up Raspberry Pi Imager, connect your microSD card writer, and install RetroPie from its Choose OS menu. Re-mount the microSD card once you've finished flashing it, because we've got some changes to make.

As with our DB9 joystick project in issue 101 (**magpi.cc/101**), we have to tell the GPIO to treat the controls as pull-up switches. Recalbox, by comparison, implements this by default.
Create the **pullup.sh** file we've supplied (**magpi.cc/pullupfix**). You can put it anywhere you like – we stuck ours in **/home/pi/**. Now open **/etc/rc.local** on the SD card and, above the `exit` line, add:

```
/home/pi/pullup.sh
```

This will load your pull-up settings on boot. If you're setting up your disk on a Linux system, you can set **pullup.sh** as executable now. Otherwise, we'll do that on first boot.

02   First boot
Make sure you have a keyboard plugged into your cabinet for this bit. We left ours propped up against the marquee acrylic during setup for easy access. A Bluetooth keyboard is a viable alternative, but it's easier to start with a wired connection.
Plug in Raspberry Pi's power. It should boot to the EmulationStation interface, but we can't configure the controls until we've set our pull-up script executable. Press **F4** to exit to the command line and type:

```
chmod /home/pi/pullup.sh +x
```

While we're here, let's enable SSH:

```
sudo raspi-config
1 system option
s3 password
```

DOWNLOAD THE FULL CODE:

magpi.cc/rpipullupfix

```
enter a new password
3 interface options
enable ssh
yes
```

You can now SSH into Raspberry Pi from another PC using a client such as Remmina or PuTTY.

03 Add hotkey button support

If, like ours, your arcade cabinet's GPIO controller setup has either one or two extra hotkey buttons for easy access to save, load, and exit shortcuts while playing, then the standard version of the mk_arcade_joystick_rpi driver available from RetroPie's package manager won't support them. We'll have to manually add an updated version from maintainer Recalbox's GitLab repo.

At the command line, type:

```
git clone --branch v0.1.9 https://gitlab.
com/recalbox/mk_arcade_joystick_rpi.git
sudo mkdir /usr/src/mk_arcade_joystick_rpi-
0.1.9/
cd mk_arcade_joystick_rpi/
sudo cp -a * /usr/src/mk_arcade_joystick_
rpi-0.1.9/
nano /usr/src/mk_arcade_joystick_rpi-0.1.9/
dkms.conf
```
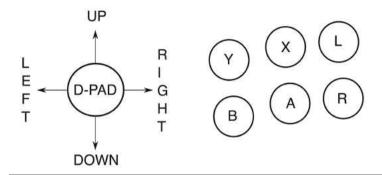
In this file, change `PACKAGE_VERSION="$MKVERSION"` to `PACKAGE_VERSION="0.1.9"`. Press **CTRL+X** to exit, then **Y** to save.
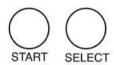
Back at the command line, enter:

```
sudo dkms build -m mk_arcade_joystick_rpi
-v 0.1.9
sudo dkms install -m mk_arcade_joystick_rpi
-v 0.1.9

reboot
```

Raspberry Pi is powerful enough to run most games, and you can stream classic and modern arcade titles from a separate PC on your network via Steam Link

Steam Link lets you create a dedicated control map. Remember to map your hotkey button to its guide button

> We'll use the RetroPie distribution to provide extra features such as Steam Link support

HackSpace 77

▲ Button and joystick correspondences for player controls; we recommend this configuration for use with RetroPie. L and R map to the right and left shoulder buttons

THE MAGPI

This tutorial is from in The MagPi, the official Raspberry Pi magazine. Each issue includes a huge variety of projects, tutorials, tips and tricks to help you get the most out of your Raspberry Pi. Find out more at **magpi.cc**

04 Optional: Load your hotkey driver

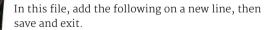When Raspberry Pi has rebooted, SSH back in and type:

```
sudo modprobe mk_arcade_joystick_rpi map=1,2
```

Go over to the arcade machine and press **F4** to get to the command line and test your controllers:

```
jstest /dev/input/js0
jstest /dev/input/js1
```

If that works, it's time to load that module on boot. At the command line:

```
sudo nano /etc/modules
```

In this file, add the following on a new line, then save and exit.

```
mk_arcade_joystick_rpi
```

Next, at the command line:

```
sudo nano /etc/modprobe.d/mk_arcade_
joystick.conf
```

In this file, add the following:

```
options mk_arcade_joystick_rpi map=1,2
```

Now save, exit and reboot.

05 Configure RetroPie

There's a bit more configuration to do before RetroPie is ready to go. SSH in and type:

```
sudo ~/RetroPie-Setup/retropie_setup.sh
```

…to open the ncurses configuration menu.

If you did not manually install a hotkey version of the mk_arcade_joystick driver in the previous steps and do not need one, go to:

```
P manage packages
driver
819 mkarcadejoystick
```

…and install it.

If you need to connect any Bluetooth keyboards or controllers, go to:

```
C Configuration / tools
804 bluetooth
```

Press **R** to register a device and follow the pairing instructions.

832 samba in the configuration menu sets up Samba shares so you can easily transfer ROMs and BIOS images over your local network

You can add extra emulators here, but we'll come to that later. For now, select the R Perform reboot option from the main menu.

06 Input configuration

When RetroPie reboots, it should inform you that it can detect two GPIO controllers. Press and hold any button on the left-hand button bank to configure controls for player 1. Because arcade controls don't map perfectly to a gamepad, you'll have to skip some buttons by pressing and holding any key.

> " When RetroPie reboots, it should inform you that it can detect two GPIO controllers "

Map up, down, left, and right on the arcade stick to the D-pad. Follow our button assignment diagram to map the top row to button Y, X, and L(eft shoulder), and the button below to buttons B, A, R(ight shoulder).

Map Start to player 1's left-hand front function button and Select to their right-hand front function button – this will be their 'insert coin'

Controller Settings

You'll probably want to reconfigure your controls in Steam Link to better match its Steam Controller-based expectations

button. In our wiring configuration, our single hotkey button – the last we set – is associated with player 1.

Approve your configuration, then set up player 2's controls in the same way.

07 Getting to know RetroPie

With your controllers configured, RetroPie's main interface will open. Press A to select menus and items and B to go back. Press Start to open the main menu and Select to open the options menu. Press the same button again to close each of these.

As you have yet to put any games on the system, only the RetroPie menu will be available. Here, you'll find easy access links to configuration tools, including some we used earlier. Install new emulators and drivers from the RetroPie Setup menu.

You'll probably need to disable overscan to get rid of a black border around the screen. In the ReotroPie menu, select Raspi-config > Display options > Underscan > No and then reboot to solve the problem. Note that button B is mapped to the **ENTER** key in this set of menus.

When you add any new games, ROMs or emulators, you'll have to restart EmulationStation by pressing Select, going to Quit, and then Restart EmulationStation.

08 Install more emulators

Although this is an arcade machine, you can play what you like on it. The core lr-mame2003 and lr-fbneo emulators are included, along with those for popular consoles

such as the Sega Mega Drive, used in some arcade systems and for which original games are legally available.

Some emulators require system BIOS images. Sadly, very few of these have been made legally available to emulation enthusiasts. SNK distributes a UniBIOS compatible BIOS set in its 40th Anniversary Collection. We recommend adding the following:

opt > 327 opentyrian – arcade-like DOS shoot-'em-up Tyrian 2.1 is now freeware.

exp > 241 lr-mame – a more up-to-date version of MAME that supports a wider range of ROMs. Install from source for bleeding edge.

exp > 307 digger – a sanctioned remaster of Windmill Software's Dig-Dug.

exp > 334 steamlink – this allows you to stream less emulation-friendly titles directly from a Steam installation on a Windows or Linux PC.

09 Configure your emulator

Once you've installed a new emulator, such as lr-mame, you'll have to configure the libretro back end to use it by default for either all games or selected titles. The easiest way to do this is to browse to the game you want to play in the EmulationStation front end.

Go to the Arcade menu, press B to start any game – it doesn't matter if it currently works or not – and then press B again when you're briefly prompted to 'press a button to configure'. Select option 1 to set the default emulator for arcade →

Top Tip 👍

Steam Link smoothly

Use a wired Ethernet connection for optimal Steam Link game streaming.

RetroPie is significantly more configurable than Recalbox, although its interface doesn't look quite as slick

games and choose lr-mame. Option 2 allows you to select a different emulator for anything that doesn't work well with this.

10 Connect Steam Link

Linking Steam to your arcade cabinet lets you stream a wealth of modern and classic arcade games to Raspberry Pi from a more powerful PC, like Melty Blood, Guilty Gear, Horizon Chase Turbo, and Street Fighter V. After you've installed it and restarted EmulationStation, go to the Ports menu and select Steam Link.

It'll download updates – you will need a keyboard plugged in to approve these – and then run. Make sure Steam is running on a PC on your local network and that Enable Remote Play is ticked under Settings > Remote Play.

On the arcade machine, select the computer you want to link to. Steam Link will show a code. Enter this in your PC's Steam client when prompted. To avoid a resolution mismatch, run Steam with a monitor that matches the resolution of your arcade machine set as your primary display.

11 Configure Steam Link

You may want to reconfigure your controls, as Steam Link doesn't inherit the control layout from RetroPie's EmulationStation, and some games do better with alternative button assignments – for example, to more closely match

an Xbox or Steam Controller, which swaps the position of the B and A buttons.

To set these, launch Steam Link, press up to highlight the gear icon, press A (per our button assignment diagram), go right to highlight Controller and press A. Select the controller you wish to configure, then press down and right twice, and select Setup Controller.

Hit the button you want to associate with each Steam Controller-style button as it's displayed on screen. Use a keyboard or your second set of controls to use the skip button at the bottom to bypass extraneous buttons.

12 Why use Steam Link?

Steam Link is an invaluable tool for arcade emulation enthusiasts, not only because you can play more CPU-intensive games, but also because it's the best way of ensuring copyright compliance for a number of re-released arcade games.

We've been playing Ghouls 'N Ghosts from the Capcom Arcade Stadium on our cabinet via Steam Link. Unlike some SNK and Sega re-releases, Capcom doesn't supply emulator-ready ROM files and the EULA for that compilation doesn't allow you to extract its PAK files.

Neil Brown of decoded.legal opines (**magpi.cc/ romextractionlegal**) that "when even a legitimate Steam purchaser extracts the ROMs and runs them on their own Raspberry Pi, they infringe Capcom's copyright", making streaming these titles your best option for fully legal home arcade action. ▫

Design your own ring

Learn to work with curves as we make
a piece of jewellery

Jo Hinchliffe

🐦 @concreted0g

Jo Hinchliffe is a
constant tinkerer and
is passionate about all
things DIY. He loves
designing and scratch-
building both model and
high-power rockets, and
releases the designs and
components as open-
source. He also has a
shed full of lathes and
milling machines and
CNC kit!

I n last month's tutorial, we looked at
projection as a tool to add geometries
or objects onto curved surfaces. In this
month's tutorial, we will discuss the limitations
of projection and look at how we can map
sketches onto the surfaces of curved objects.
Along the way, we will make a geeky maker ring
to 3D-print, and we will also explore some more
complex uses of the Curves Workbench.

Last month we projected objects onto curved
surfaces, and this works well in many instances. It
has some limitations, though; the major one is that by
design, projection works with a fixed projection view
and as such, it can't be used to wrap a design around

Figure 1 ◈
An example of mapping sketches around cylinders, this 'maker
chic' 3D-printed ring acts as a spanner for M2 to M5 nuts!

another design. Let's start by looking at how we can
wrap a sketch around a cylindrical object.

Open FreeCAD and use the Addon Manager to add
the Curves Workbench. Click Tools > Addon Manager.
When the list populates, scroll to the Curves
Workbench and click to install. Once downloaded and
installed, you will be prompted to restart FreeCAD.

Once you have restarted FreeCAD, open a new
project. For our first example, let's start in the Part
Workbench. Click the 'Create a cylinder' tool icon
and then highlight the 'cylinder' object in the file tree.
In its dialog box, change the radius to 10 mm and
the height to 30 mm. Next, let's switch over to the
newly installed Curves Workbench. In the preview
window, select the surface that forms the outer wall
of our cylinder, and then click the 'Map a sketch on
a surface' tool icon. You'll notice in the file tree view
that a new 'SketchOnSurface' object appears and
that this is a drop-down menu item. If you click it to
open the drop-down menu, it should contain only one
item: mapped sketch. If you double-click on this item,
it will open the Sketcher Workbench. In the sketch,
there will be a blue rectangle. When geometry lines
in sketches in the Sketcher Workbench are blue, it
means that they are construction geometry that acts
as a guide but doesn't appear in the sketch as part
of the object. They have many uses, but in this case,
the 'Map a sketch on a surface' tool has created a
construction geometry that matches the dimensions
of the cylinder's face and acts as a containing area
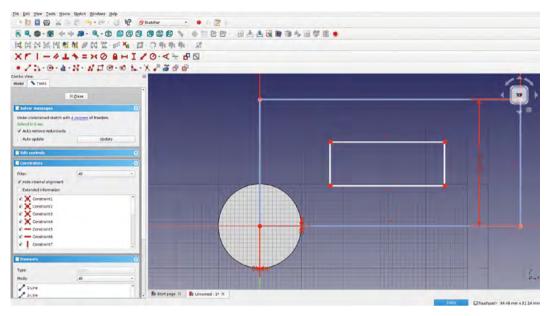for our mapped sketch. For a first simple example,

let's draw a rectangle somewhere inside the blue construction rectangle (**Figure 2**).
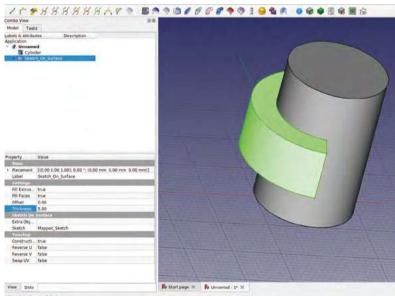
Having drawn a rectangle inside the construction geometry, close the sketch and you should automatically return to the Curves Workbench. You should now see the rectangle we drew on the surface of the cylinder in relative position when the blue construction geometry is wrapped around the cylinder. If you don't see your mapped rectangle, rotate the cylinder as it may be positioned out of view. You can now close the drop-down mapped sketch on the file tree and when you select the parent SketchOnSurface item, the small mapped rectangle should be green and selected in the preview window. Selecting this, we can perform some operations on this sketch like any other. With the sketch highlighted in the dialog, set the 'Fill faces' option to 'true' and then increase the 'thickness' a few millimetres. You'll see the sketch is extruded out from the face of our cylinder (**Figure 3**).

It's perhaps worth pointing out that the resulting objects made using mapped sketches work in the same way as other objects with regard to the Part Workbench and other tools. In **Figure 4** (overleaf), you can see that we have added some fillets to the edges of the part and used the appearance menu to set the object's colour. We've also gone one step further in that we have clicked the surface of our mapped object and created another SketchOnSurface object to create the small extruded hexagon.

As well as adding more mapped sketches to build up complex geometries, we can use Part Workbench tools such as the Boolean operation type tools to make cut-out parts using the mapped sketch. As an example, we thought of a fun mini project to make and 3D-print. We are going to make

a useful (ish!) 3D-printed ring. Let's start in a new project and once again with the Part Workbench. Create a cylinder that is just a fraction larger than the diameter of one of your fingers and around 12 mm tall.

Create another cylinder that is slightly larger in diameter but the same height – the difference between the first and second cylinder diameter will be the thickness of the ring. If you don't move either cylinder, they should be aligned and concentric around the origin point. Select the second larger cylinder first and then select the first cylinder and click the 'Make a cut of two shapes' tool icon from the Boolean tools selection. You should now have a basic ring shape. →

YOU'LL NEED

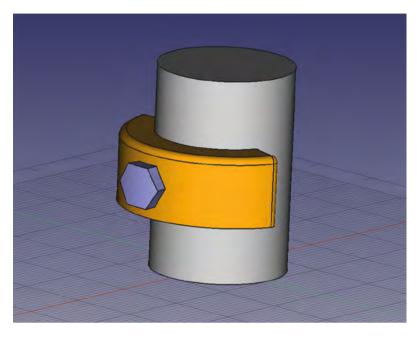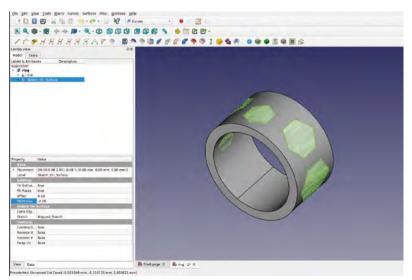A computer and a copy of the 0.19 stable version of FreeCAD

Figure 4 ◈
Parts created using
mapped sketches
behave similarly to
other objects, and you
can add fillets and
chamfers and change
their appearance
settings as usual

Figure 4 ◈
Parts created using
mapped sketches
behave similarly to
other objects, and you
can add fillets and
chamfers and change
their appearance
settings as usual

Figure 5 ◈
Our mapped sketch
laid out and partially
constrained with
the M-series nut
dimensions, plus a small
clearance allowance

Figure 6 ◈
Our hexagonal spanners extruded through the ring using
a negative thickness value

Moving to the Curves Workbench, we will map a sketch to the outside surface of the ring as we did earlier with our cylinder. Click the outer surface and then click the 'Map a sketch on a surface' tool icon once again. Click to drop-down and double-click the mapped sketch, which will open in the Sketcher Workbench.

Our idea for this maker ring is that we will cut out some hexagonal shapes that not only look decorative but will serve as an emergency small spanner tool. As such, inside our construction geometry, we added some hexagons sized to common small metric nut sizes. We went for M2, M2.5, M3, M4, and M5. You don't need to fully constrain the sketch, but we opted to align one line of the hex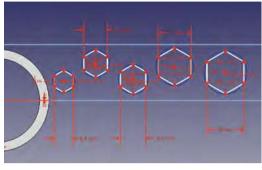agon with a vertical constraint so that we could then constrain the nut size by selecting two opposite points and applying a horizontal dimension. We added roughly 0.1 mm to the M-series given nut sizes to make them an easy fit – you can see the sizes we used in the constraint values (**Figure 5**). We preferred our hexagon spanners positioned randomly, but you could constrain them to be on the centre line and more equally spaced if preferred.

Once you are happy with your hexagons, close the sketch and you should see the mapped sketch around our ring. Select the SketchOnSurface object and, in the dialog, set 'fill faces' to 'true' – but this time, add a negative value equal to the thickness of your ring design in the 'thickness' input. Our ring is 2 mm thick, so we went with -2 mm. This should now have extruded the mapped sketch through the ring (**Figure 6**).

Now that we have all the objects created, it's a case of moving to the Part Workbench and selecting the ring first, then the SketchOnSurface object, and then creating a cut of the two parts to create our ring. We can then use File > Export to save our ring as an STL file suitable for slicing and printing. You can see the ring we made in **Figure 1**.

Using the sketch mapping tools on the Curves Workbench is just one example of what can be done with this workbench. Almost every tool on the Curves Workbench could have a tutorial written about it, but let's look at a couple of tools to make a complex curved surface that curves in all axes.

We are aiming to make an object that looks like an aeroplane cockpit canopy cover as it's a challenging shape to model. Again, we are not particular about the dimensions of the object and, as such, we are not worried about constraining aspects of the design.
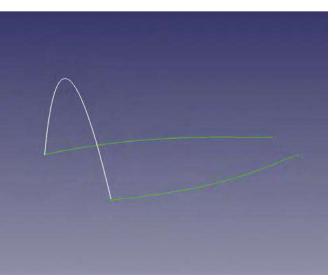
To begin, create a new project, move to the Part Design Workbench and create a body. Create a sketch in the XY plane; in this sketch, let's create a B-spline. This B-spli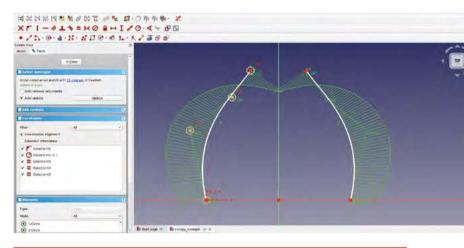ne is going to be the lower side edge profile of our canopy cover. Click the B-spline tool and start the B-spline attached to the X axis to the left of the origin point. Add some points and then finish the B-spline to make it look similar to the left-hand side of **Figure 7**. Before we close the sketch, select the B-spline that we just made and click the 'Create a symmetric geometry with respect to the last selected line or point' tool to add a mirrored B-spline on the right-hand side of the Y axis line (**Figure 7**).

Create another sketch in the active body, but this time in the XZ plane. We need to import the endpoints of the mirrored B-splines that sit on the XY plane (the starting end of the B-splines in the previous sketch). In the new XZ plane sketch, click the 'Create an edge linked to an external geometry' tool and zoom in to select the endpoints of the two B-splines from the first sketch. Now that we have those points available, draw another B-spline that begins with the point constrained to the left-hand B-spline from the first sketch. Place the second B-spline point on the Z axis line above the origin and attach the final B-spline point to the imported edge of the mirrored B-spline. →

Figure 7 ◆
Using the 'Create a symmetric geometry' tool on the Sketcher Workbench to create two B-splines that will form the side edges of our canopy

Figure 8 ◆
A B-spline in an XZ plane sketch attaches the rear edges of the original B-splines and will form the rear edge of the canopy
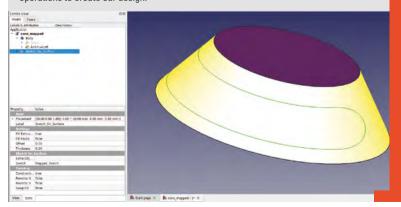

CURVED CONE

If you want to map a sketch onto a cone shape and you want to work in Part Design, the 'Map a sketch to a surface' tool works a little differently. You have to draw your own construction geometry and do things in a slightly different order. As an example, in a new project, move to the Part Design Workbench. Create a new body and create a first sketch in the XY plane. Create a circle roughly 100 mm in diameter around the origin point. Close this sketch. Create a second sketch in the XY plane and draw a smaller circle for the top of our cone around the origin point – again, don't worry about constraining these sketches too much. Closing the second sketch to return to Part Design, highlight the second sketch and in the dialog, under the 'Attachment' submenu, go to 'Position' and raise the Z axis value – we went with 25 mm, to move the sketch upwards.

Select both the sketches in the file tree and then click the yellow and red 'Loft a selected profile through other profile sections' tool icon. Click OK and you should have a cone. Moving to the Curves Workbench, select the curved outer face of the cone and then click the 'Map a selected sketch' button. Different to earlier examples, you will see that a mapped sketch object appears without the SketchOnSurface object. Double-click the mapped sketch object and you will move to the Sketcher Workbench, but there won't be any construction geometry.

Toggle the 'Toggles the toolbar or selected geometry to/from construction mode' button to turn the sketching tools blue, and then draw a construction geometry rectangle from the origin point. You can work out the exact shape and size of the curved cone surface limits, but for this example we just drew a rectangle that would fit on the cone surface with room to spare.
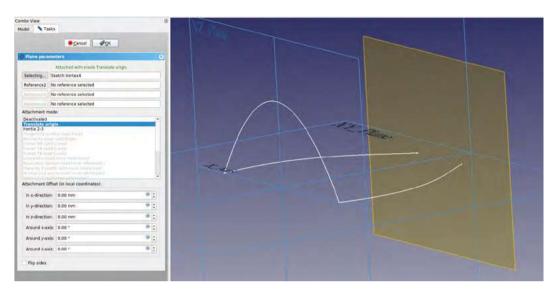
Toggle the Sketcher tools back so that they no longer draw geometry objects, and draw an object inside the rectangle. We went for a 'slot' object. Close the sketch to return to the Curves Workbench. Finally, select both the 'additive loft' object that is our cone and the mapped sketch in the file tree and then click the 'Map a selected sketch' button again. It should now create the SketchOnSurface object and map our slot onto the cone's surface. Similar to earlier examples, we can add thicknesses and perform cut operations to create our design.

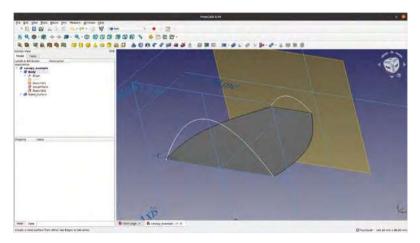> Creating datum planes is an incredibly useful **tool in many projects**

Figure 9
Setting up a new datum plane to attach a sketch to

Figure 10
Creating a ruled surface is the first step of creating our final curvy canopy

When you finish this B-spline, you should have a symmetrical curve that will form the rear edge of our canopy design (**Figure 8**).

We now need to add a similar smaller B-spline that will connect the front ends of the original mirrored B-splines to form the front edge of our canopy design. Of course, these points don't sit on a datum plane, so we need to add a datum plane to work on first. Creating datum planes is an incredibly useful tool in many projects. To help us set up our new datum plane, let's make the three standard plane

origins visible. To do this in the file tree, highlight 'origins?' and press the **SPACE** bar. You should now see the blue lines and labels for the XY, XZ, and YZ planes. Click the 'Create a new datum plane' tool icon on the Part Design Workbench. You should see a 'Plane Parameters' dialog box appear in the Combo view area. At the top of the dialog, you will see four boxes with 'No reference selected' written in each box. To the left of the first box, the button text should read 'Selecting' – if for any reason it doesn't, click on that box until it does. If we imagine our desired new datum plane by visualising where we want to sketch, we could describe the plane as a sort of XZ plane moved to the far end of our original mirrored B-splines. This kind of hints as to how we will create the new datum plane. First, let's click on the XZ plane that we made visible in the preview window. You should now see that the new brown-coloured datum plane moves onto the XZ plane. Double-check the dialog; you should find that the first box now has 'XY_plane' in the input, and the button now says 'Plane'. Click the button until it returns to 'Selecting'. Finally, we need to move and zoom to select the endpoint of one of the mirrored B-splines we made in the original sketch. This should make that vertex point the reference of our new datum plane, which stays oriented as parallel to the XZ plane. Once it's in place, you can click OK to close the datum plane dialog (**Figure 9**).
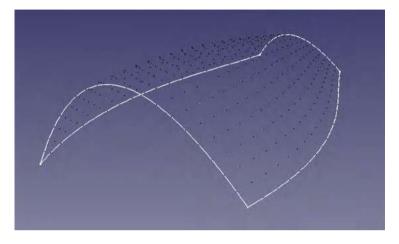
With the new datum plane highlighted in the file tree, click the 'Create a new sketch' tool to open a new sketch. Import the B-spline endpoints to the sketch and make a smaller B-spline to form the front of the canopy in the same way that we did the sketch

for the canopy rear on the XY plane.

To create a surface for our canopy, first we are going to make a flat ruled surface between the original sketch lines and then we are going to use the 'Sweep profiles on 2 rails' tool and the 'Approximate points to NURBS curve or surface' tool. To create the ruled surface, move to the Part Workbench, select the two B-splines of the original sketch, and click the 'Create a ruled surface' tool (**Figure 10**).

Moving to the Curves Workbench, in the file tree view, select the new ruled surface object and select the two sketches that create the rear and front edge shape of our canopy design. With these three objects highlighted, click the 'Sweep profiles on 2 rails' tool. Your computer may take a little time to process this, but once it's complete, you should have a collection of points, or a 'point cloud' in the preview window that is beginning to look like the shape of our aeroplane canopy (**Figure 11**). There are plenty of options to increase and decrease the resolution of the point cloud, but for this example the default settings are adequate.

To convert the point cloud into a surface, make sure that the Sweep2Rails object in the file tree view is highlighted, and then click the 'Approximate points

to NURBS curve or surface' tool. You should now have a perfect canopy surface. Of course, the surface created is similar to our earlier ruled surface in that it has no thickness dimension – we can, however, use familiar tools to create a thicker solid object. With the 'Approximation Curve' object highlighted in the file tree, you can move to the Part Workbench and click the 'Offset' tool. In the offset tool dialog, you can create an offset and check the 'fill offset' box to create a thicker canopy (**Figure 12**).

We've seen a small range of tools on the Curves Workbench in this tutorial, but we've only scratched the surface of what this workbench is capable of. As ever, there is a heap of knowledge on the FreeCAD community forum as well as lots of videos and tutorials online to help you explore the rest. ▫

Figure 11
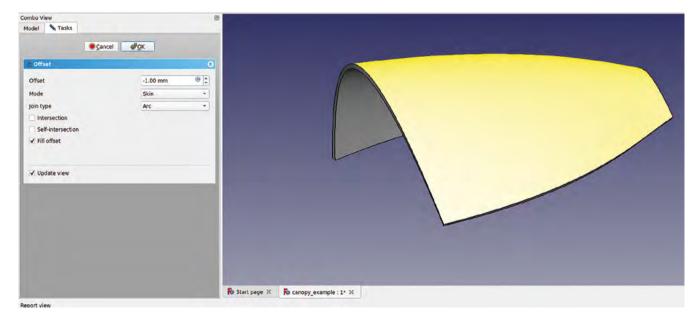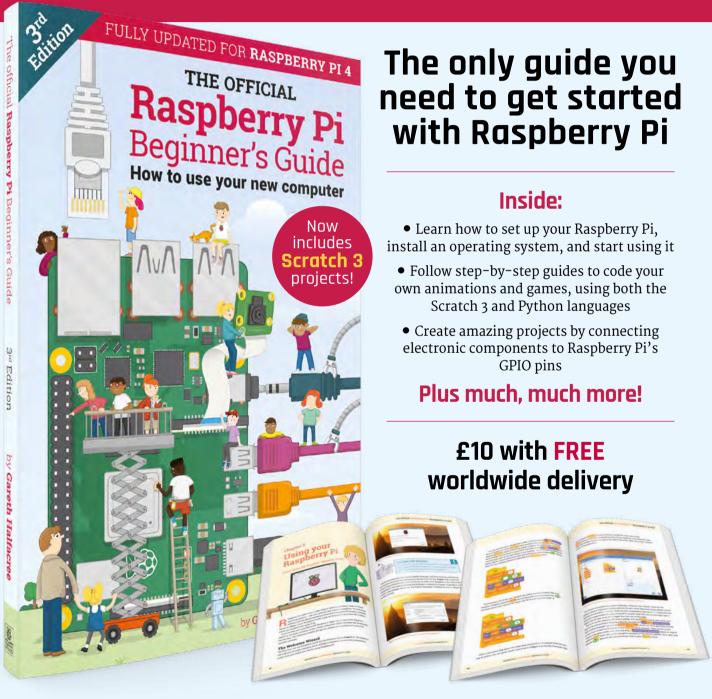Creating a point cloud which we can convert into a curved surface using the Curves Workbench

Figure 12
Our completed canopy design finished by creating a filled offset

Build a Makerspace for Young People

Join our **free online training course on makerspace design** to get expert advice for setting up a makerspace in your school or community.
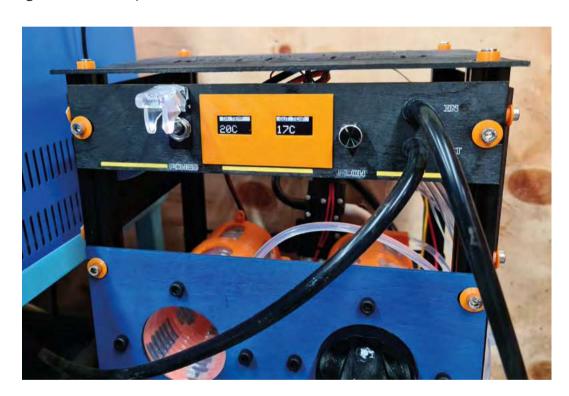
Sign up today: rpf.io/makerspace

3D-print a cooler for your laser cutter

Use recycled beverage cans to cool your laser cutter

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Laser cutters generate a lot of heat, and if that heat isn't carried away quickly enough, it can permanently damage your laser tube. This project lets you use your existing kitchen-based cooling infrastructure (your fridge-freezer) to cool your kit with some recycled beverage cans.

If you don't supply your laser with pure water to cool it, then you're going to have a bad time. That's why your laser tube is designed with a water jacket. If you're a home hobbyist living in a moderate climate and you only use your laser occasionally, then you'll probably find that a pump and a large water reservoir is enough for your needs. However, if the temperatures rise and your need for the laser increases, you'll find that a water tank alone gets too hot to effectively cool the laser for any length of time.

OVERHEATING IS BAD

There are lots of different solutions for active water cooling involving radiators, Peltier cells, or dedicated industrial chillers. Radiators and fans are fine if your workshop environment is cool enough, but if it's too hot outside, the radiator will work in reverse and heat up the water in the coolant tank. Peltier cells work, but they're extremely inefficient and will need a hefty power supply to support them. Industrial chilling units are the preferred solution, but the cost of them is higher than the initial cost of the laser cutter, so for the casual hobbyist, they're probably out of the question.

In an ideal world, throwing some ice or a cooling block from the freezer into the cooling tank of your laser would be enough to keep the water cool, but in reality, this would be a disaster. The voltages

Above
The panels can be mounted vertically or horizontally – whichever orientation is most convenient will work as long as the cooling cans are sealed properly

used inside the laser tube are so high that they can overcome the resistance of the glass, and pass through into coolant. If the coolant is polluted by minerals from tap water or metal salts, then the laser tube can arc through the metal chassis, through the pipework, through the next-door neighbour's cat, and through anything else that it's in contact with.

This project finds a middle-ground solution, protecting the purity of the coolant, but using an existing refrigeration system to chill things down in a way that means you don't need to have a high-end industrial chiller attached to your laser to use it in hot weather. It's also a scalable solution to some extent, in that increasing the cooling capacity just means adding extra coils to the system, and increasing the usable time means adding more coolant cans to the freezer.

STAY PURE

Surely distilled water isn't dangerous? Well, no, it isn't really dangerous in normal use. However, it isn't as inert as you might think that it is either. Long-term exposure of metals like stainless steel, brass, and copper to distilled water leads to leaching of mineral salts, corrosion, pitting, and other unpleasant effects. When working with distilled water, it's best to stick with plastic if you want to keep things as pure and clean as possible.

The cooler is a simple heat exchanger, pulling water from the laser's coolant tank and passing it through refrigerated coils. Your first job is to print the coil housings and populate them with plastic tubing. You should aim to leave about 15 cm of tubing sticking out from each end of the coil housing; you can hold the coils in place using hot glue on the outside of the housing.

You need to make two connectors that will take the pipes from the cooling coils and connect them to the input/output hoses that lead to the cooling tank. The internal diameter of the coils is 4 mm, →

Below
If getting the plastic tube to wind into a coil is tricky, try wrapping it around a piece of metal tube and annealing it with a propane torch or heat gun

YOU'LL NEED

- **Arduino Uno** (or similar)
- **2 × 100 kΩ NTC thermistor**
- **2 × 100 kΩ ¼ watt resistors**
- **2 × 0.97" monochrome OLED screens** (SSD1306 type)
- **4LPM 12 V diaphragm pump**
- **SPST toggle switch with 12 mm shaft**
- **1 × 10 A PWM motor speed controller module** (optional)
- **12 V PSU** (at least 3 A)
- **4 × 700 mm 2020 aluminium extrusion**
- **20 × M5 roll-in spring T-nuts**
- **20 × M5 10 mm cap head machine screws**
- **7 × A4 sheets of 3 mm plywood or acrylic**
- **5 m of 6 mm ID black silicone tubing**
- **5 m of 6 mm OD 4 mm ID tubing**
- **10 × barbed drip pipe connector 4 mm ID**
- **4 × barbed drip pipe connector 4 mm and 8 mm ID, 4-way irrigation system**

Making sure that the thermistor is waterproof before you fit it into the barbed connector is important

Below ◈
The schematic shows the connections to the Arduino and power supply. Note that the optional motor speed controller is not shown in this diagram. If you are using the speed controller, attach it to the pump after the toggle switch

and the input/output pipes are 8 mm. Connecting two coils together in parallel using a four-way barbed drip pipe connector will give you a good flow-rate through your pump. Each pair of coils fixes onto a laser-cut mounting panel, which in turn bolts into place on a rack-mount case made from 2020 aluminium extrusion.

TOO COOL IS UNCOOL

You will also want to add a thermistor to these connectors, so you can monitor the temperature at the input and output of the cooler. To make each connector, close off one end of a 4 mm/8 mm barbed connector by melting it and squashing it against a hard object. This should leave you with a connector
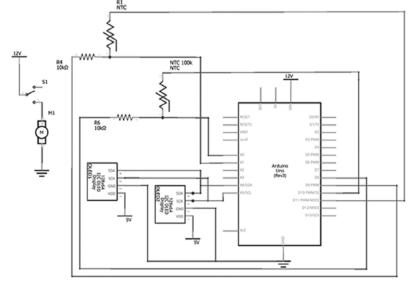
that has one 8 mm barb and two 4 mm barbs. Join this connector to a second 4 mm/8 mm barbed connector using a short piece of 8 mm pipe. Close off one of the 4 mm barbs on the second connector by melting it closed, or filling it with silicone or hot glue. Insert the thermistor into the remaining 4 mm barb and seal it into place using silicone, but first you should apply a thin layer of silicone over the thermistor and let it set, so that the thermistor is waterproof.
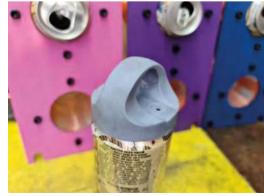
The thermistors connect to an Arduino that displays the temperature on OLED screens. The monitoring of the temperatures is passive, although it is possible to control the pump actively using the Arduino's PWM pins and a TIP122, or similar transistor, if you want to modify the code. The Arduino and all electronics fit onto another laser-cut control panel, and are held in place with screws and hot glue. Connect the electronics as shown in the diagram, and upload the sketch.

The outlets for the input and output hoses are also cut into the control panel, and can be held in place with hot glue and cable ties. The positioning of the pump will vary depending on the position of your current cooling setup, and it can be held in place on the chassis using cable ties to reduce vibration. The pump is connected so that it pulls water from the cooling system, forces it around the coils, and the outlet pipe returns the water into the coolant tank.

The rack-mount chassis for the project is very simple to make. It uses two A4-sized pieces of

Below ↗
You should now have a basic cooling unit with a number of slots for coolant cans, but no coolant cans. The cans are very easy to make. Fill an empty 500 ml beverage can with anti-freeze, and seal the top with some gaffer tape. 3D-print the can handles, and fit them to the top using plenty of silicone rubber. For extra security, use self-drilling metal screws to screw through holes in the handle and into the top of the can. Once the silicone rubber is dry, pop the cans into the freezer until you need them

plywood for the top and bottom, and two laser-cut pieces for the side handles. To mount the top and bottom of the chassis, use an M5 tap to thread the centre holes in each piece of 2020 extrusion, then drill the top and bottom pieces with a 6 mm drill, and bolt through from the top and bottom. The hole positioning for the top and bottom pieces can be determined by using the other laser-cut panels as templates. Connect the other panels to the 2020 extrusion, and mount the top and bottom accordingly. The holes should be 26 cm apart side to side, and 16 cm front to back.

THAT'S SO COOL

There are a few improvements you can make to the system to improve its functionality. Make some dummy cans out of insulating foam, and use these to fill the holes where the cans go when the system isn't in use. The insulating foam means that you can run the system with less than a full panel of cans, and you won't waste as much energy cooling the free air in the empty coils. The dummy cans will also block light from getting into the system, reducing the risk of an algal bloom in the tank. Insulating the outside of the coils is also a very good idea to

improve efficiency, so wrap bubble-wrap or foam insulation around the outside of the coils, and hold it in place with tape or wire.

Improve the accuracy of the temperature readings by following Adafruit's guidance – switch the Arduino's AREF to 3.3 V, adjust the resistor in the potential divider circuit on the thermistors to optimise the temperature range you're sampling, and take multiple samples to get an average temperature rather than a single point sample. ▫

CAREFUL POSITIONING

Wouldn't it be better to connect the cooler to the existing coolant pump and cool the water immediately before it goes into the laser? Although the idea of using the existing coolant pump is tempting, it means the cooler can't be removed easily from the coolant tank without risking an airlock in the cooling system, and pumping very cold water directly into the laser is a very bad idea. Although the laser needs to be cooled, excessive cooling can cause thermal shock, which can crack the laser tube. Imagine dropping iced water into a pickling jar that's just come out of the oven – the sudden cold shock would shatter the jar.

Finish your projects

Let's turn those piles of bits into working builds

Ben Everard

🐦 @ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Let's be honest: there's a good chance that you have a half-finished project sitting on your shelf, or in a box right now. And there's a good chance that it's accompanied by several more half-finished projects. Sometimes it feels like unfinished projects are the curse of the maker. But do they need to be?

Is it possible to be a maker and actually finish a significant percentage of your projects?

We ran an entirely unscientific poll on Twitter to find out why people aren't finishing their projects. The answers are:

1. Lack of time (43%)
2. Lost interest (29.1%)
3. Deciding how to make it (19.9%)
4. Waiting on parts (7.3%)

Almost half the makers who answered don't finish their projects because they don't have enough time. It's fully understandable – we have more than a few unfinished projects around HackSpace towers waiting for time to finish them. However, is there a way to avoid this project-killer?

We cannot get you more time to work on your projects. We all have multiple pulls on our limited number of hours in the day, and we can't magic that away, no matter how much we'd like to. However, that doesn't mean that there's nothing you can do to help you when projects stall due to lack of time.

Here are a few of our key tips to making sure you don't run out of time on your projects.

SIMPLIFY, SIMPLIFY, SIMPLIFY

Yes, we know that project you saw on YouTube/Instagram/Facebook/Twitter (delete as appropriate) had so many bells and whistles that you wanted to add to your project. However, just because someone else has had time to complete a complex mammoth of a project, doesn't mean you do.

What is the smallest set of features you need for your project to work? That's what you need to do to finish it.

Remember – a finished project is one that works (in whatever sense that means for you). If it's worthwhile, there will be an opportunity for a version 2 with added features later.

You can frame this as having an 'exit plan'. Your project may need features A and B. If you have time, you can add features C and D, but if you run out of time, you can exit after feature B and still call your project done.

Silicon Valley people like to call the simplest possible version of a project the 'minimum viable product'. That is, the smallest set of features you need for your project to be useful. If it is useful (or whatever other metric you want to use to judge your project), then you'll find time to add the extra features.

BUDGET

Many of the points regarding time also apply to budget. Simplifying a project also usually makes it cheaper. Having features that you can skip if you don't have time also means there are features you can skip if you run out of budget.

Using parts and techniques that you are familiar with means that you can properly plan the cost better. Using anything unfamiliar increases the risk of finding that you need extra parts or tools that you didn't initially budget for.

Ultimately, it comes down to picking the right project. At the start, you need to be realistic about the time and money you have available, and make sure that your project fits into both.

MORE THAN ONE PROJECT?

This is tied into the previous point. Sometimes the best thing you can do is split something up into multiple smaller projects so that each is useful in its own way. You can complete the first one, then have a working project while you start work on the second one. You may decide that the second one is unnecessary, or want to do it in a different way after finishing the first one.

PROJECTICIDE

The quickest way to finish a project is to put it in the bin. Figuratively of course – reuse and recycle what you can. This might sound brutal, but an unfinished project takes up mental load, and that takes time away from other projects.

Sometimes projects are good to start, and you learn as you go. At a certain point, the project may no longer be useful. If so, it's time to kill it.

It might be hard to kill a project that you've invested time and money in. After all, once you've invested a certain amount, it'd just be wasteful not to finish it, right? Business people call this the sunk cost fallacy. This means that the only relevant information you need when deciding whether or not to finish a project

> **Any project that you enjoy or learn from is a successful** project, whether or not it ever reaches its final form

is the amount of time and money it will take to finish it. The amount of time and money you've already spent is (in this decision) irrelevant – it's gone and it's not coming back, whether or not you finish the project.
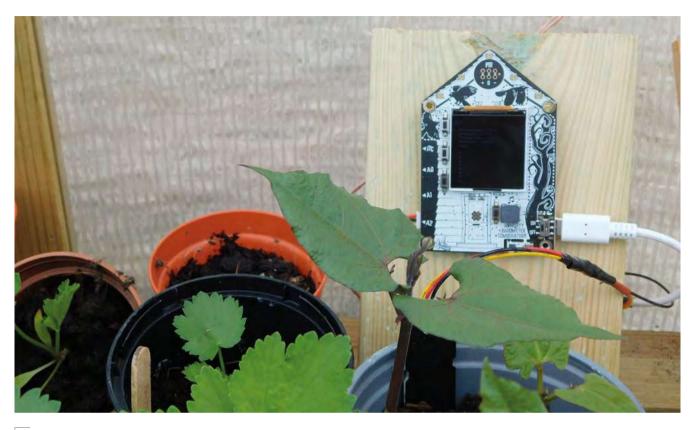
Killing a project doesn't mean that the project has failed. We make projects for many reasons besides getting the end goal – to learn, for fun, for curiosity. Often, we achieve these goals before the project is finished, and that's why the project never gets finished. Any project that you enjoy or learn from is a successful project, whether or not it ever reaches its final form.

If you kill any projects you're not going to finish, this will free up space in your workshop, budget, and brain for those that are still worth finishing. Be ruthless. →

MINIMISE UNKNOWNS

If a project is going to take a while, stick to things you know as much as possible. Use microcontrollers and other components that you're already familiar with. If a new one is technically better, but will take time to learn, then only use it if you absolutely need the functionality it has. The same is true for techniques. An imperfect technique that you're familiar with may be better than a superior technique that you've not used before.

IGNORE ALL OF THIS
IF IT'S NOT USEFUL

People make things for all sorts of reasons, and we're not here to tell you the right way to make a project, because there is no universal right way. The important thing is that you're doing it in a way that's right for you.

If you like having a shelf full of unfinished projects that you can tinker with at your leisure, then go right ahead and continue working that way. If you enjoy having an endless project that will probably never be finished, but is a source of constant tinkering, then again, continue doing this.

However, if you get frustrated by never getting to the end of your projects, we hope these tips are useful for you.

Similarly, you're often in the position where you can do something from scratch or you can use a pre-built module to simplify the build. Be realistic about the amount of time and effort you have to put into the project, and use whatever you can to reduce the risk of getting bogged down in unnecessary complexity.

This doesn't mean you should get stuck in a rut of never exploring new things. Here at HackSpace towers, we like to separate exploring new tools, techniques, and components with projects that we want to get done. If you just want to experiment with something new, do it, but that doesn't mean you have to make a big project with it.

Make time for playing and exploring that's not about a finishing goal. Once you've had a bit of a play with them, you can plan to incorporate them into a project, but by then you'll have a good idea of the time and effort it will take.

DO IT

Yes, this sounds obvious, but if you want to finish a project, you have to actually work on it when you do have some spare time.

Working on a project is a habit. It can be a habit you're in and you head straight to it when you've got a bit of time. If it's a habit you're out of, you'll find yourself thinking that you should be doing it
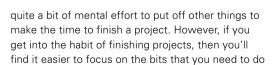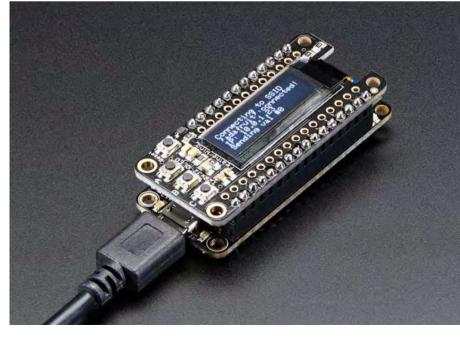
A FINISHED PROJECT
IS A GOOD PROJECT

It's common to feel some form of imposter syndrome in making. Many projects require working across a range of disciplines, and it's usual to not be an expert in all of them. It's easy to look at other projects and want to be an electrical engineer, a mechanical engineer, a programmer, a woodworker, and an artist. However, you're (probably) not all these things. Nobody is. A good project can use pre-built modules rather than home-designed circuits. It can use off-the-shelf mechanics; it can use code straight from example files. It can be a jumbled mess of glue and wires. It can be housed in an old ice cream tub rather than a custom-designed enclosure. If it works for you, then it's a good project.

when you're sat doom-scrolling on social media, or channel-hopping. If you're out of the habit, then it can take a bit of effort to get back into the habit of working on it when you have free time, but once you make it a habit, it'll become second nature.

Finishing projects is also a habit. If you find that you rarely finish projects, then you probably won't feel the urge to finish the next project and it can take

> **The order in which you do things can have a big** influence on whether or not you finish a project

quite a bit of mental effort to put off other things to make the time to finish a project. However, if you get into the habit of finishing projects, then you'll find it easier to focus on the bits that you need to do to finish the project.

PLANNING
The order in which you do things can have a big influence on whether or not you finish a project. There are a few things to bear in mind when planning a project:

- **Fail fast** – If your project has a bit that may be difficult, or something that may turn out to be impossible (or beyond your time, material, and expense budget), you want to find that out as soon as possible.

- **Don't do all the fun bits first** – All projects have fun bits and boring bits. If you do all the fun bits at the start, then you only have the boring bits left and there's a good chance that the project will die.

- **Identify the easy wins** – These are things that only take a bit of time, but are enjoyable and feel like you're making a lot of progress for little effort. When you're struggling for motivation, do an easy win. When you've got time, do some of the harder work. □

Left
This author has more than one project happily running in an old ice cream tub. It may not look great, but it's quick, cheap, and it works

Below
Plug-in modules

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated
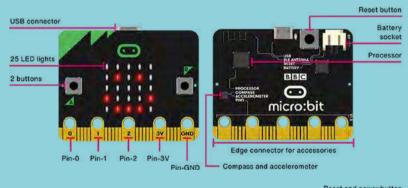
Making music with the micro:bit

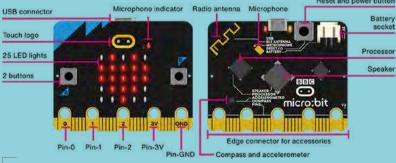Accessories that allow you to express your inner musical self

By **Marc de Vinck**   @devinck

t's been over five years since the BBC, in collaboration with a few dozen partners, launched the Make It Digital initiative. From that program, they created the micro:bit, the open-source electronics learning tool that was distributed to over a million students in the UK by 2016. Around the same time, the Micro:bit Educational Foundation was created, which expanded the mission statement to include all students from around the world. It's been a great success which has been embraced by the DIY electronics community, further expanding its popularity.

The original micro:bit didn't include a speaker, and although making music with the board was possible, the first thing you would need to do is add some kind of speaker. The latest version of the board adds the ability to detect sound, and includes a speaker and a few other sensors and features. It also adds a capacitive touch sensor on the front. All these additions make creating music a lot easier right on the board. In this Best of Breed, we'll be looking at expanding the musical capabilities of the micro:bit through add-on boards. If you want to learn more about this affordable little ARM-based processor, check out the official site at **microbit.org**.

Above ◈
The top two images show either side of the original micro:bit, with the new improved version shown below

Kitronik :KLEF Piano for the micro:bit vs noise:bit

KITRONIK ◈ **$21.97** | **kitronik.co.uk**　　　**PIMORONI** ◈ **$15.52** | **pimoroni.com**

What's not to love about the simplicity and beauty of the Kitronik :KLEF Piano for the micro:bit? The classic grand piano-shaped PCB not only looks great, it allows you to compose monophonic music through its 15 capacitive touchpads. Thirteen of the pads are arranged as a single octave, and the remaining two touchpads allow you to shift octaves up and down. There is also an on-board amplifier and speaker, allowing for a louder playing experience. The board also features a USB plug which passes the power to the micro:bit, so you don't have to power it separately. Head over to the Kitronik website to learn more and download the demo code.

Pimoroni says, **"Make your micro:bit sing!"** And with the noise:bit, you can. Make your original micro:bit come to life with the on-board TPA301 amplifier and 8 Ω 2 W speaker. It's also useful with the v2 of the micro:bit, since it has a larger speaker and amplifier.

> Advanced users can use MicroPython to import the **speech module to make the micro:bit talk**

You can use the noise:bit with the block-based Microsoft MakeCode editor and the included audio generation blocks. You can make it beep, boop, and bleep. And more advanced users can use MicroPython to import the speech module to make the micro:bit talk. →

VERDICT

Kitronik :KLEF Piano

What's not to love in this form factor?

10/10

noise:bit

A simple way to add sound to an original micro:bit.

9/10

Launchpad Breakout Board for micro:bit

ADAFRUIT ◈ **$14.95** | adafruit.com

T he Launchpad Breakout Board for micro:bit is an add-on board that breaks out the majority of the pins on your micro:bit, or Adafruit CLUE, so you can easily use alligator clips, conductive fabric, or even conductive paint with your board.

The standard form factor of the micro:bit is nice, but accessing all the individual pins is difficult. Most beginners only use the pins on the board that allow for alligator clips to be attached. But if you look closely, there are a lot more pins available, and that's

> If you look closely, there are a lot more pins available, **and that's where this board comes into play**

where this board comes into play. With it, you will have easy access to not only multiple 3.3 V and ground connections, but also pins 0, 1, 2, 8, 13–16, 19, and 20. That should give you enough access to make an interesting musical instrument that has multiple analogue and digital inputs.

Left ◈
All the pins are on easy-to-access pads

VERDICT

Launchpad Breakout Board for micro:bit

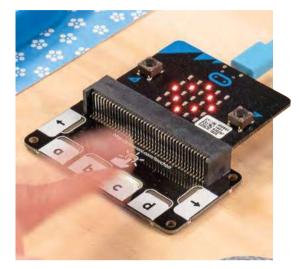**Perfect for crocodile clips.**

9 /10

touch:bit

PIMORONI ◈ **$12.42** | pimoroni.com

The touch:bit from Pimoroni is **another quick and simple way to get six touch-sensitive buttons and LEDs added to your micro:bit.** What we like about this, as opposed to touching the pins directly, is the size of the pads. You get a lot more room to press a specific button, and there aren't a bunch of wires or alligator clips to get in the way.

There are also six bright white LEDs, one at each key, which adds a little bit of fun, or feedback, depending on your program. It's a bit limiting, since you only get six pads, and they aren't all in a row allowing you to play it like a mini piano, but we can imagine it being very useful for triggering sounds or other music-related code that can be activated via a digital button.

VERDICT

touch:bit

A simple way to add an interface to your micro:bit.

8/10

Speaker for micro:bit

PIMORONI ◈ **$9.66** | pimoroni.com

Above ◰
Tap, tap, tap, tap

Below ◿
Clip on a speaker

If you own an original version **of the micro:bit, this is a must-have accessory.** It's hard to make music without a speaker! The Speaker for micro:bit from Pimoroni uses alligator clips to connect directly to your micro:bit. No soldering required. It's amplified, which helps boost the sound, just don't expect room-filling sound with deep bass! Pimoroni has a HEX file available to download on the product page to get you up playing a song quickly. They also go over the basics for creating your own music via the popular Blocks Editor and Python. →

VERDICT

Speaker for micro:bit

A nice addition for an original micro:bit.

8/10

Bit:Commander Console & Controller for micro:bit

PIMORONI ◆ **$18.29** | pimoroni.com

I **know exactly what you are thinking.** This looks like a gaming device, not a musical instrument. And you're right! But think about all the fun you can have with the Bit:Commander Console & Controller when it comes to controlling sound. The board features a built-in speaker, a dial, joystick, multiple buttons, and a row of multicolour LEDs.

The Bit:Commander Console is a great way to have both analogue and digital inputs connected together, allowing you to quickly create a unique musical instrument. And since it's a classic game controller-like layout, it should be easy to use and feel familiar. There is no soldering required, and the creators have provided a package that you can install in MakeCode to make it simple to program and harness all those features. Check out the link for more information. ◻

Left ◆
Lights? Check.
Buttons? Check.
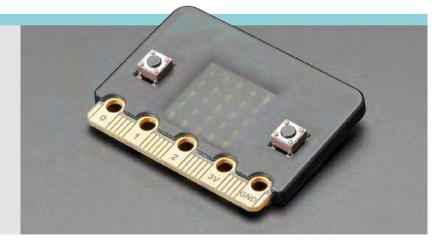Music? Check

VERDICT

Bit:Commander
Console
& Controller

A great interface for gaming and music.

10/10

KEEPING SAFE

ADAFRUIT ◆ **$2.95** | adafruit.com

When you are making anything with the micro:bit, especially something that is handheld, you are more than likely going to want some kind of protection for the board itself. There are a lot of different cases available, but we really like this Smoke Snap-on Case. It feels and looks good. It also allows for the LEDs to shine through when lit, but hides them when they are not in use. It's a cool effect, and a great case.
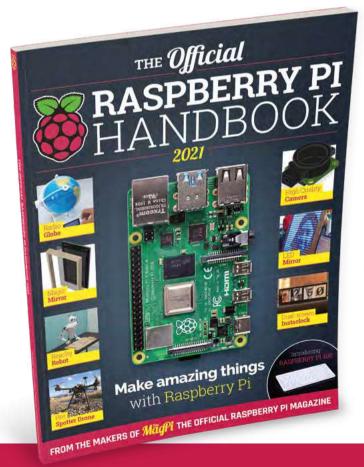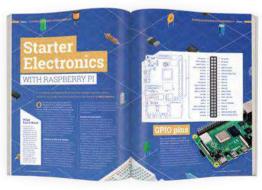
Jay-D

The maker microcontroller for mixing music

CIRCUITMESS ◈ **£71.98** | **circuitmess.com**

By **Ben Everard** 🐦 **@ben_everard**

Jay-D is a DJ mixtable. Basically, it enables you to control two songs at once – speed them up, slow them down, add effects, and control the volume of each independently. This means you can create a seamless blend of two songs.

There are two slide potentiometers at the top that control the volume of each of the two tracks, and one in the middle that controls the balance between the two. Each side also has three rotary encoders for controlling various effects that you can apply, such as the speed, high/low pass filters, and bitcrusher. The interface is reasonably intuitive, but there's also a guide at **hsmag.cc/JayDGuide** to help you get started. Really, the challenge isn't controlling the device, but understanding how to use it to make music that sounds good – a trait that Jay-D has in common with most musical instruments.

Take a look here for a mix put together by AA Battery to get an idea of what Jay-D can do: **hsmag.cc/JayDMix**.

Right ◈
With seven knobs and three sliders, this is a very twiddly interface

Jay-D arrives as a kit of parts. The surface mount bits are all pre-soldered onto the circuit boards, but you have to solder on the larger through-hole bits and assemble the acrylic case. It's all quite straightforward and would be a good project for a beginner. In fact, Jay-D started life as part of CircuitMess's STEM Box for helping people understand how to use technology.

With everything assembled, it's time to play some tunes! Jay-D comes with a microSD card pre-loaded with some songs to try out. We'd love to give you some advice here, but we're not particularly gifted in the art of DJing here at HackSpace towers, so we'll leave it to you to work out what you like.

There are a few limitations with the firmware. The biggest is that while you can load two songs and mix them in different ways, you can't load a third song while the music is playing. This means you can use it to create mixes of songs, but not create longer mixes of three or more songs.

PROGRAMMING

As with all of CircuitMess's creations, Jay-D isn't just an end-product for you to use – it's a programmable device for you to tinker with and tweak. It's based on ESP-32, and you can use it in all sorts of ways (for example, there's a Bluetooth MIDI controller library for ESP-32 that we're keen to try out with Jay-D).

However, for most people, the programming is probably going to be using the Arduino Library for Jay-D (**hsmag.cc/JayDLibrary**) or CircuitBlocks (the visual programming tool provided by CircuitMess). At the moment, there's only documentation on the latter (see **hsmag.cc/JayDCode**). CircuitBlocks does show the Arduino code for the program you're creating, so it can be useful to first try building something in there to see how the library works, even if you want to use a more traditional Arduino setup.

Using either method, you can control the four LED matrices and apply different effects to songs and mix them in different ways. All the basic features of playing audio, applying effects, and mixing together two tracks are handled by the library, so you can get started with this very quickly.

We've had fun playing with Jay-D. It's particularly useful that the kit comes with a range of different songs so that you can use it without first having to try to source some songs.

This is an unusual device, and how much enjoyment you get from it really depends on how you like to interact with music. It takes a bit of getting used to, and the results aren't for everyone. It's unlike anything else we've seen in the maker scene (there are more fully featured DJ tables available, though they're not as hackable as Jay-D). If you enjoy programming and want to experiment with different ways of mixing and manipulating songs, then there's loads to play with here. □

VERDICT

A hackable DJ table for mixing songs and applying effects.

8/10

Add 57,600 pixels to your Raspberry Pi Pico

Can this turn the Pico into a games machine?

SPOTPEAR ⬥ $11.89 | hsmag.cc/picoscreen

By **Ben Everard**   🐦 **@ben_everard**

The snappily named Raspberry Pi Pico display 1.54-inch LCD by Spotpear brings in a 240×240 pixel IPS screen and ten buttons in a joypad-like arrangement. There's four for direction, four for action, a select, and a start. At least, they're labelled like this. You can use them for anything you like.
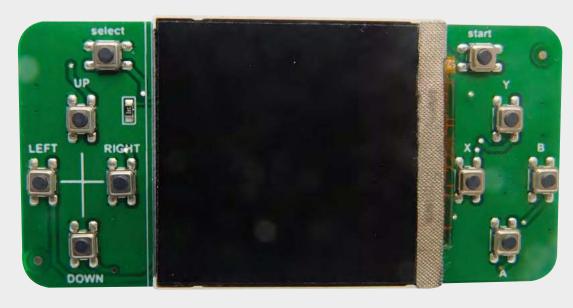
To help you get started, there's a short manual at **hsmag.cc/SpotpearGuide**, which includes example code for MicroPython and C.

This example code is easy enough to use, but it is a little messy. The mechanism for controlling the hardware isn't separated into its own module, so you're left with either the task of building the library yourself or having slightly untidy code. Not the biggest inconvenience, but compared to how neatly some maker hardware companies manage their code, we found ourselves off to a disappointing start.

There are also some sample UF2 files included along with the C example code, but these appear to have been built for different hardware and work either partially or not at all. The actual example code did compile and work properly.

When we ran the example code, we were impressed with the quality of the screen. With 240×240 pixels in just 1.54 inches, there's a high pixel density that can give crisp graphics. Obviously, high

Right ⬥
The buttons are just a bit too small and fiddly for us

pixel densities are a double-edged sword. While they can look great, it does mean higher RAM use, more time transferring data, and more data to process.

Fortunately, Pico is well-suited to the task of driving screens. Each pixel can take 16 bits of colour data, so a full-frame buffer is just 115,200 bytes. The display data is transferred by SPI, and Pico has a maximum SPI frequency of half the clock speed. For MicroPython, that means 62.5MHz. The actual data transfer rate is a little less than this because of overhead of the protocol, but we were able to drive full-frame refreshes at over 40 fps, which is plenty for smooth animations.

Obviously, if you're looking to do animations, sending the data is only half the story. You also need to calculate the frame before it's displayed. If you're using MicroPython, you are quite limited by the amount of processing you can do and still keep a high frame rate (though you could use the second core to offload some of the processing). With C, you've got much more scope, especially as you could potentially offload the data transfer using direct memory access (DMA).

The one disappointing thing about the screen is that there's no control over the backlight. According to the documentation, it should be attached to pin 13, but it isn't. You can't turn it on or off – it's just permanently on, and quite bright. That's a deal-breaker for anything

> **They're the sort of buttons that are commonly used as reset buttons as they're hard to accidentally press**

running off battery power, as it will suck up a lot of power. However, if you want a display permanently on, this might be perfectly acceptable.

While we were quite impressed by the screen, we can't say the same for the other part of the hardware – the buttons. They're small, stiff, and have very little movement. The end result is a button that is hard to press, and hard to know if you've pressed it. They're the sort of buttons that are commonly used as reset buttons as they're hard to accidentally press.

We had hoped that this screen would make a good base for a games console, but unfortunately these buttons would just make for a frustrating experience. They might be OK for a menu-driven user interface, but that's about it.

Another minor annoyance in this is the lack of any mounting holes. This makes it hard to embed into a project as the user interface.

We wanted to like this project. It's got a good, high-res screen and a nice layout of buttons. However, the choice of components makes it hard to see how we'll use this in our projects. We're considering removing the surface-mount buttons and soldering wires onto them to make a more useful device, but if you're going to go to that level of surgery, it's probably better to start with a plain screen and work your way up from there. ◻
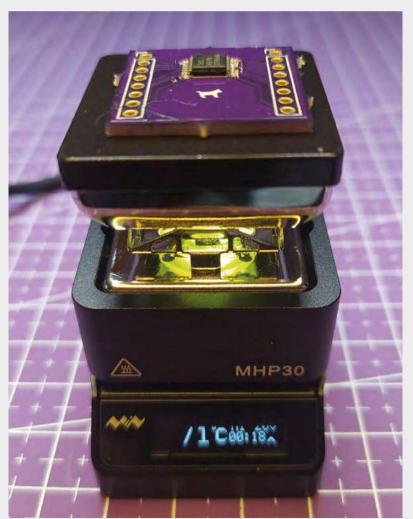
VERDICT

Good screen, but awful buttons.

5/10

Miniware MHP30 mini hot plate preheater

Solder small boards or fry quails' eggs

MINIWARE ◆ **£75** | miniware.com.cn

By **Jo Hinchliffe** 🐦 **@concreted0g**

The name Miniware may sound familiar, as it is the company that produces the excellent TS100 and TS80 soldering irons. Building on those products, they have recently released a mini hot plate, the MHP30, suitable for populating small single-sided SMD boards. After seeing the MHP30 featured in Adafruit videos and elsewhere in the community, we decided to grab one and take a look.

Available for around £75, the MHP30 arrives in excellent packaging that is good enough to be used as a storage case for the device when not in use. Inside the box, the MHP30 has a silicone cover over the hot plate, and tucked away in the box base is a rather lovely one-metre-long silicone USB-C to USB-C cable.

It really is tiny – the hot plate is 30 × 30 mm – but, for such a tiny device, it feels very well built in hand and it's incredibly cute. It's reasonably heavy for its size, and feels well-placed on a desk and unlikely to topple in operation. However, for extra security there are two rotating legs that increase the stability when deployed. We also noticed that there is a threaded M3 hole with which, we imagine, you could attach the device to a larger homemade base, if you desired. For power, it requires a USB-C PD charger, common these days as laptop power supplies. It needs a reasonable amount of power, and worked well with our laptop supply that can supply 20 V at 3 A. For curiosity, we also tried it with a USB PD power bank that could supply up to 15 V at 1.5 A, and this didn't work.

Power the MHP30 on and the small OLED display

Left ◆
Having successfully reflowed a powerpad-style IC, the hot plate is cooling and the RGB LED shows the temperature

springs into life. If you have used the TS100 or TS80 irons, then this firmware will feel similar, although not exactly the same. On the back of the machine there are two buttons, and you quickly get used to how they interact with the menu items. If you press the button on the left-hand side (as viewed from the front of the device), the device immediately starts to heat towards the first target temperature. There are three editable temperature settings in memory, and it's easy to switch between them. The unit can heat up to 350°C, and we set ours to 220°C as we had some solder paste that would reflow at just under that temperature. Looking through the menu and the instruction manual, there's a range of sensible adjustable parameters and sleep modes to tinker with. It's important to note though that the device doesn't do curved or programmable cycles through different temperatures, although you could set three temperatures in memory and quickly switch between them.

When powered up, you'll notice that the area underneath the hot plate lights green – this is an on-board RGB LED that changes colour with the temperature, acting as a visual reminder of when the plate is hot. It moves from green to amber to white, to red eventually, and also repeats this pattern as it cools. Green is reserved for when the hot plate is less than 50°C, which is the temperature where it is pretty safe to handle. We'd recommend that you keep the unit powered on whilst cooling, as when powered off there is no way to tell if the unit is still extremely hot.

We loaded an ATtiny IC onto a small breakout PCB, with some solder paste applied to the pads. We set our fume extractor/ filter up near the device and, placing the PCB on the hot plate, we set the machine

to heat to 220°C. Within a couple of minutes, the paste flowed perfectly and a good result was achieved. A long press of the right-hand button kicks the machine out of heating mode and back into the general startup menu, and the plate begins to cool. It takes longer than you would imagine to cool down, possibly as we are used to dealing with small-tipped soldering irons. We also soldered a board with a powerpad-style IC on it, where there is a pad underneath the chip that would be impossible to solder with a soldering iron. Again, the MHP30 performed really well.

It's hard to find fault with the MHP30 – it is a well-put-together little device. At 30 mm square, it may be too small for some applications, but it can also accommodate larger work that hangs off the plate edge a little. We found the menu system straightforward, and the small OLED display easy to read. We love the RGB LED that acts as a visual clue to the temperature, and we find the device overall very well thought out and easy to use. ◻

VERDICT

A super-cute and usable tool for SMD work at home.

9/10

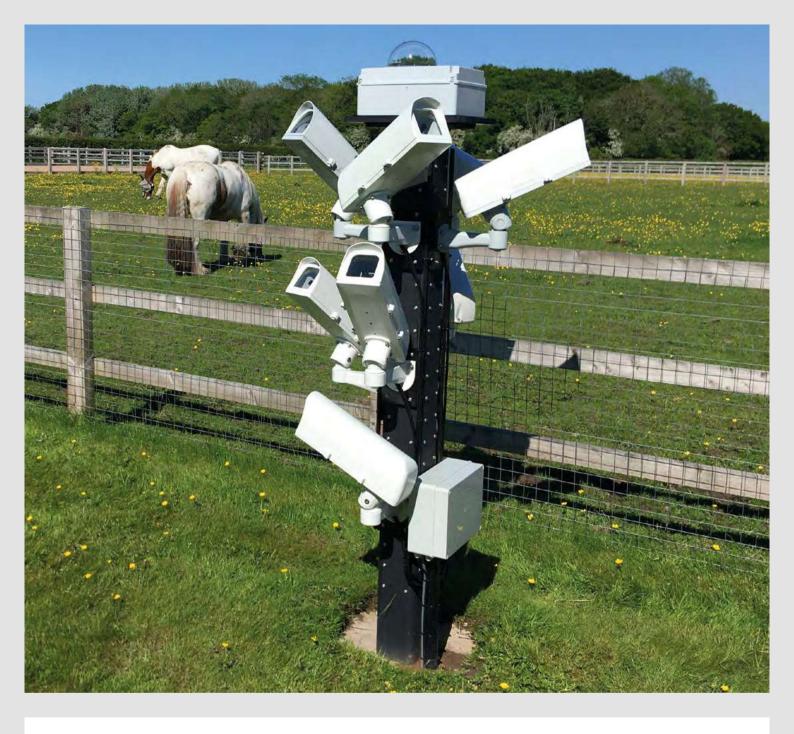## issue #47

ON SALE
23 SEPTEMBER

—

3D PRINTING

—

ALSO

→ **RASPBERRY PI PICO**

→ **GAMES CONTROLLERS**

→ **VINYL CUTTING**

→ **HOT AIR**

→ **AND MUCH MORE**
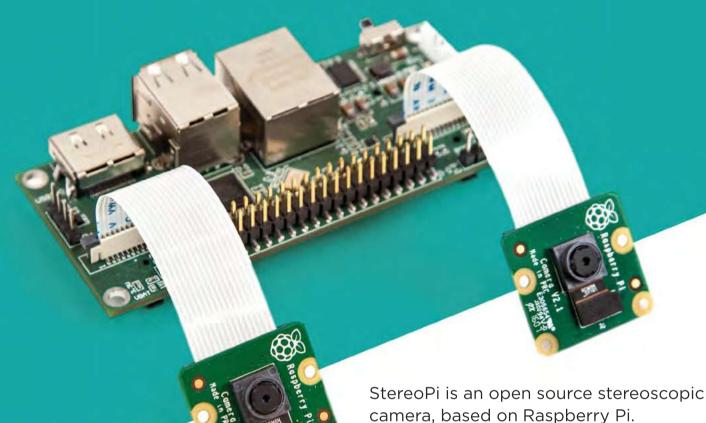
—

DON'T MISS OUT

hsmag.cc/subscribe

Did you know that every day, hundreds of tons of small interplanetary objects enter Earth's atmosphere? These tiny, typically pea-sized fragments can offer us a unique view on the formation and evolution of our solar system. Yet, most of them remain unobserved.

The Global Meteor Network's mission is to observe these meteors, aka shooting stars, using a global network of cameras pointed at the night sky. Each camera is connected to a Raspberry Pi running open-source software for video capture, compression, and meteor detection. Currently, the project has over 200 camera stations based in 20 countries, with data published daily on the project website.

Do you know
HOW ROBOTS SEE?

StereoPi is an open source stereoscopic camera, based on Raspberry Pi.

This is the friendliest way to play with the stereoscopic vision for the spatially aware robots, 3D photo and video!

RASPBERRY PI INSIDE

STOCK RASPBIAN SUPPORT

Raspbian

OPEN SOURCE

CROWDFUNDED PROJECT

LinuxGizmos.com
"The StereoPi can capture, save, livestream, and process real-time stereoscopic video and images for robotics, AR/VR, computer vision, drone instrumentation, and panoramic video."

MickMake
"With it you can do things like, stream stereoscopic 3D video to YouTube, build real-time depth maps using OpenCV, create panoramics using Hugin and even a 3rd person view of real life. Cool."

Raspberry Pi Blog
"There are some excellent community efforts too, of which our current favourite is this nifty dual camera board."

Hackster News
"You can hook this up to YouTube, to Oculus Go, you can use it with OpenCV.. I cannot wait to start messing around with these because it's basically a dream come true."